

Docket No. 214295US2RD/btm



217#5

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

IN RE APPLICATION OF: Mutsumi FUJIHARA, et al.

GAU: 2171

SERIAL NO: 09/960,337

EXAMINER:

FILED: September 24, 2001

FOR: DATA MANAGEMENT METHOD AND STORAGE MEDIUM STORING DATA MANAGEMENT PROGRAM

REQUEST FOR PRIORITY

ASSISTANT COMMISSIONER FOR PATENTS
WASHINGTON, D.C. 20231

RECEIVED
NOV 28 2001
Technology Center 2100

SIR:

- ☐ Full benefit of the filing date of U.S. Application Serial Number [US App No], filed [US App Dt], is claimed pursuant to the provisions of 35 U.S.C. §120.
- ☐ Full benefit of the filing date of U.S. Provisional Application Serial Number, filed, is claimed pursuant to the provisions of 35 U.S.C. §119(e).
- ☒ Applicants claim any right to priority from any earlier filed applications to which they may be entitled pursuant to the provisions of 35 U.S.C. §119, as noted below.

In the matter of the above-identified application for patent, notice is hereby given that the applicants claim as priority:

<u>COUNTRY</u>	<u>APPLICATION NUMBER</u>	<u>MONTH/DAY/YEAR</u>
JAPAN	2000-332993	October 31, 2000

Certified copies of the corresponding Convention Application(s)

- ☒ are submitted herewith
- ☐ will be submitted prior to payment of the Final Fee
- ☐ were filed in prior application Serial No. filed
- ☐ were submitted to the International Bureau in PCT Application Number .
Receipt of the certified copies by the International Bureau in a timely manner under PCT Rule 17.1(a) has been acknowledged as evidenced by the attached PCT/IB/304.
- ☐ (A) Application Serial No.(s) were filed in prior application Serial No. filed ; and
(B) Application Serial No.(s)
 - ☐ are submitted herewith
 - ☐ will be submitted prior to payment of the Final Fee

Respectfully Submitted,

OBLON, SPIVAK, McCLELLAND,
MAIER & NEUSTADT, P.C.

Marvin J. Spivak
Registration No. 24,913

Surinder Sachar
Registration No. 34,423



22850

Tel. (703) 413-3000
Fax. (703) 413-2220
(OSMMN 10/98)

09/960,337

日 本 国 特 許 庁
JAPAN PATENT OFFICE



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2000年10月31日

出 願 番 号

Application Number:

特願2000-332993

出 願 人

Applicant(s):

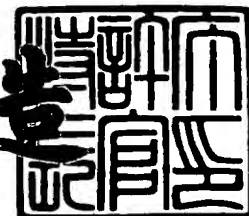
株式会社東芝

RECEIVED
NOV 28 2001
Technology Center 2100

2001年 9月19日

特 許 庁 長 官
Commissioner,
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3086199

【書類名】 特許願

【整理番号】 12830001

【提出日】 平成12年10月31日

【あて先】 特許庁長官殿

【国際特許分類】 G06F 5/00
G06F 7/22

【発明の名称】 データ管理方法、データ検索方法、コンピュータ読み取り可能な記録媒体、データ管理装置及びデータ検索装置

【請求項の数】 18

【発明者】

【住所又は居所】 神奈川県川崎市幸区小向東芝町1 株式会社東芝 研究開発センター内

【氏名】 藤 原 睦

【発明者】

【住所又は居所】 東京都府中市東芝町1番地 株式会社東芝 府中事業所内

【氏名】 斉 藤 悦 生

【特許出願人】

【識別番号】 000003078

【住所又は居所】 神奈川県川崎市幸区堀川町72番地

【氏名又は名称】 株式会社 東 芝

【代理人】

【識別番号】 100064285

【弁理士】

【氏名又は名称】 佐 藤 一 雄

【選任した代理人】

【識別番号】 100088889

【弁理士】

【氏名又は名称】 橘 谷 英 俊

【選任した代理人】

【識別番号】 100082991

【弁理士】

【氏名又は名称】 佐 藤 泰 和

【選任した代理人】

【識別番号】 100108062

【弁理士】

【氏名又は名称】 日 向 寺 雅 彦

【手数料の表示】

【予納台帳番号】 004444

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 データ管理方法、データ検索方法、コンピュータ読み取り可能な記録媒体、データ管理装置及びデータ検索装置

【特許請求の範囲】

【請求項 1】

管理すべき複数のデータを複数のデータ格納領域に別々に格納し、前記複数のデータ格納領域に格納されたデータ間の関連性に関する情報を前記複数のデータ格納領域あるいはこれら以外のデータ格納領域のそれぞれに付随した関連性情報可能領域の少なくともいずれかに格納することを特徴とするデータ管理方法。

【請求項 2】

前記関連性情報格納領域に格納される前記情報は、その関連性情報格納領域が付随したデータ格納領域とは異なる他のデータ格納領域を指定する情報を含むことを特徴とする請求項 1 記載のデータ管理方法。

【請求項 3】

前記他のデータ格納領域を指定する情報は、データ格納領域のポインタであることを特徴とする請求項 2 記載のデータ管理方法。

【請求項 4】

前記管理すべき複数のデータのいずれか 2 つ以上が同一である場合に、これら 2 つ以上の同一のデータをひとつのデータ格納領域に格納することを特徴とする請求項 2 または 3 に記載のデータ管理方法。

【請求項 5】

前記同一のデータが格納されたデータ格納領域が、他のデータ格納領域に付随する関連性情報格納領域において 2 回以上指定される場合があることを特徴とする請求項 4 記載のデータ管理方法。

【請求項 6】

検索パターンを識別するためのデータをデータ格納領域に格納し、このデータ格納領域に付随した関連性情報格納領域には、前記検索パターンのソーティング順序でソーティングされた前記情報を格納することを特徴とする請求項 2 ～ 5 のいずれか 1 つに記載のデータ管理方法。

【請求項 7】

前記管理すべき複数のデータを更新した後に、前記他のデータ格納領域を指定する情報を更新されたデータに基づいて再びソーティングされた情報を前記関連性情報格納領域に格納することを特徴とする請求項 6 記載のデータ管理方法。

【請求項 8】

前記ソーティング順序は、前記他の複数のデータ格納領域を指定する前記情報の配列順序をキー順位としたものであることを特徴とする請求項 6 または 7 に記載のデータ管理方法。

【請求項 9】

前記ソーティングされた情報の管理を、スプレイ・ツリー・アルゴリズムにより行うことを特徴とする請求項 6 ～ 8 のいずれか 1 つに記載のデータ管理方法。

【請求項 10】

前記関連性情報格納領域に格納される前記情報は、その関連性情報格納領域が付随したデータ格納領域を指定する情報を含むことを特徴とする請求項 2 ～ 9 のいずれか 1 つに記載のデータ管理方法。

【請求項 11】

複数の前記データ格納領域をひとつのデータ集団名に対応させて管理することを特徴とする請求項 1 ～ 10 のいずれか 1 つに記載のデータ管理方法。

【請求項 12】

前記管理すべき複数のデータの関連を表すために、前記データ格納領域を指定する情報の集合体である指定情報配列を形成し、

前記指定情報配列の少なくとも一部を前記関連性情報格納領域に格納することを特徴とする請求項 2 ～ 11 のいずれか 1 つに記載のデータ管理方法。

【請求項 13】

前記関連性情報格納領域に格納された情報から前記指定情報配列を辿ることが可能なことを特徴とする請求項 12 記載のデータ管理方法。

【請求項 14】

管理すべき複数のデータが複数のデータ格納領域に別々に格納され、前記複数のデータ格納領域以外のデータ格納領域に付随した関連性情報可能領域に前記複

数のデータ格納領域のそれぞれを指定する情報が所定の順序で格納されてなるデータベースから所定のデータを検索する検索方法であって、

前記関連性情報格納領域のうちで、検索条件と合致した順序で前記情報が格納されている関連性情報格納領域を検索するステップと、

前記検索された前記関連性情報格納領域に格納されている前記情報のうちで、検索条件に合致した前記情報を検索するステップと、

前記情報が指定する前記データ格納領域に格納されたデータを読み出すステップと、

を備えたことを特徴とするデータ検索方法。

【請求項 1 5】

データを管理するプログラムを格納したコンピュータ読み取り可能な記録媒体であって、前記プログラムは、

管理すべき複数のデータを前記コンピュータに付随した記憶装置に設けた複数のデータ格納領域に別々に格納させるステップと、

前記複数のデータ格納領域に格納させたデータ間の関連性に関する情報を前記複数のデータ格納領域あるいはこれら以外のデータ格納領域のそれぞれに付随して設けた関連性情報格納領域の少なくともいずれかに格納させるステップと、

を備えたことを特徴とする記録媒体。

【請求項 1 6】

データを検索するプログラムを格納したコンピュータ読み取り可能な記録媒体であって、前記プログラムは、

管理すべき複数のデータが複数のデータ格納領域に別々に格納され、前記複数のデータ格納領域以外のデータ格納領域に付随した関連性情報可能領域に前記複数のデータ格納領域のそれぞれを指定する情報が所定の順序で格納されてなるデータベースにおける前記関連性情報格納領域のうちで、検索条件と合致した順序で前記情報が格納されている関連性情報格納領域を検索するステップと、

前記検索された前記関連性情報格納領域に格納されている前記情報のうちで、検索条件に合致した前記情報を検索するステップと、

前記情報が指定する前記データ格納領域に格納されたデータを読み出すステッ

プと、

を備えたことを特徴とする記録媒体。

【請求項 1 7】

データ入力部と、データ処理制御部と、データ記憶部と、を備えたデータ管理装置であって、

前記データ入力部は、管理すべき複数のデータを入力し、

前記データ処理制御部は、前記管理すべき複数のデータを前記データ記憶部に設けた複数のデータ格納領域に別々に格納し、前記複数のデータ格納領域に格納したデータ間の関連性に関する情報を前記複数のデータ格納領域あるいはこれら以外のデータ格納領域のそれぞれに付随して前記データ記憶部に設けた関連性情報格納領域の少なくともいずれかに格納する、ことを特徴とするデータ管理装置。

【請求項 1 8】

データ入力部と、データ処理制御部と、データ記憶部と、を備えたデータ検索装置であって、

前記データ入力部は、検索条件を入力し、

前記データ記憶部には、管理すべき複数のデータが複数のデータ格納領域に別々に格納され、前記複数のデータ格納領域以外のデータ格納領域に付随した関連性情報可能領域に前記複数のデータ格納領域のそれぞれを指定する情報が所定の順序で格納され、

前記データ処理制御部は、

前記データ記憶部が有する前記関連性情報格納領域のうちで、検索条件と合致した順序で前記情報が格納されている関連性情報格納領域を検索するステップと、

前記検索された前記関連性情報格納領域に格納されている前記情報のうちで、検索条件に合致した前記情報を検索するステップと、

前記情報が指定する前記データ格納領域に格納されたデータを読み出すステップと、を実行する、ことを特徴とするデータ検索装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、データ管理方法、データ検索方法、コンピュータ読み取り可能な記録媒体、データ管理装置及びデータ検索装置に関し、さらに詳細には、計算機などの主記憶上に構成され、あるいは計算機などにおいて実行可能な各種アプリケーションシステムにおいて用いるのに好適なデータ管理方法、データ検索方法、コンピュータ読み取り可能な記録媒体、データ管理装置及びデータ検索装置に関する。

【0002】

【従来技術】

計算機を用いて実行されるアプリケーションシステムにおいて、データベースは通信と並んで中核技術に位置づけられ、特に性能を決定する重要な要因となっている。現在、「データベース」といえば、OODB (Object Oriented DataBase) など一部では利用されているが、通常はRDB (Relational DataBase) をさすことが多い。

【0003】

【発明が解決しようとする課題】

しかし、プラント制御システムに代表される大規模で性能要件の厳しいシステムにおいては、RDBは以下のような問題を有するために、実際には、ほとんど効果的に利用されていない。

【0004】

(1) 性能要件

性能要件の厳しいシステムにおいては、例えば、異常発生時などのように通常の10倍から100倍のトランザクションが短時間に発生するような場合にあっても、要求される厳しい性能要件を満足しなければならない。しかしながら、RDBでは最悪値の保証が非常に難しいために、どのくらいのシステム構成にすればどのような条件でも性能要件を満足できるか確定させることができず、実用に耐えない。

【0005】

また、RDBではハードディスク装置上にデータを管理することが前提であるため、たとえフルキャッシュにしてデータをすべてメモリ上に持ってくることができたとしても、データ処理の性能向上には限界が有り、実際、あらゆるチューニング手段をとっても要求性能を実現することができない場合も多い。

【 0 0 0 6 】

(2) データ構造

RDBは、表（テーブル）形式のデータモデルを採用するため、このモデルに合わないデータ構造、例えばネットワーク型データ構造を扱うことが苦手である。ネットワーク型データ構造をRDBで扱おうとすると、非常に複雑なテーブル構成をとらざるを得ず、その結果として性能が劣化する。

【 0 0 0 7 】

また、このような複雑なデータをアプリケーションで扱う場合には、アプリケーションで用いるデータ構造はネットワーク型であるのに対して、データベース側では表データでこれを表現しているため、両者の間のデータ構造が大幅に異なる。つまり、アプリケーション側において、データベースのデータ構造をアプリケーションのデータ構造へ非常に面倒な手続きを介して変換する必要が有り、データアクセス処理が煩雑でアプリケーションの開発生産性や保守性を大幅に劣化させる。

【 0 0 0 8 】

(3) 構造変更の容易性

システムのライフサイクル中に機能の改造や拡張が発生することはよくあることだが、システム設計時には予想もつかなかったような機能の追加が必要とされることがある。そのような場合に、当初のデータ構造では対応できない状況が起きてしまい、それに対応するためのデータベースの変更を繰り返すことにより「つぎはぎ」の構造になりがちである。このようなデータ構造になると保守性が著しく下がってしまい、生産性や性能が劣化してしまう。

【 0 0 0 9 】

(4) 大規模システムの開発方法論

アプリケーション依存のデータ構造実現が難しい。すなわち、大規模システム

の開発に際しては、同時期に多くの開発者がアプリケーションの開発に携わる必要がある。それぞれの担当範囲で扱うデータ構造は、データベースで管理しているデータ構造のごく一部に限られるが、データ構造は多くの場合アプリケーションについての情報無しに単独で設計されるため、アプリケーション開発者にとってはとても使いにくいデータ構造になることが多い。大規模開発のためのデータ構造開発方法を含む開発スタイルがないため、後戻りや無駄な処理が発生し、生産性が悪くなってしまう。

【0010】

以上、列挙した問題は、制御系システムに限ったことではなく、昨今流行のインターネット／イントラネットのシステムでも同じような状況を引き起こす。

【0011】

これらの問題は、主として、RDBを始めとする従来のDBMS (DataBase Management System) が、「スキーマ」と呼ばれるデータ管理の枠組みを使っていることに起因する。スキーマは、予めデータ構造の枠組みを決めておき、データを、「データ項目」と呼ばれるその枠組みに付与された意味に基づいて、登録、更新、削除、検索を行うものである。RDBにおいて、登録されるデータレコードは「タプル」と呼ばれるが、スキーマに基づくデータ管理上の制約から、タプルを一意に識別するための識別子となるデータ項目が必要になる。そして、登録データにそのような性質が無い場合には、新たに識別の為のデータ項目を付与し、何らかの方法によって識別データを作成し元のデータに付与した上でデータベースへ登録する必要がある。

【0012】

この制約のために、タプルを識別するだけで他に何の意味もないデータが付与され、データサイズが増えてしまう。また、スキーマがデータの意味を与え、これがアプリケーションでの処理とは無関係に決められるために、同じデータなのに異なるデータとして定義されたりし、無駄なデータや無意味なデータが増えてしまう。

【0013】

以上説明したように、これらの問題は全てスキーマに起因して生じるものであ

る。

【 0 0 1 4 】

本発明は、以上詳述した課題の認識に基づいてなされたものであり、その目的は、データ処理性能を向上させ、各種のデータ構造に対する適応性が高く、データ構造の変更も容易で、大規模システムの開発も容易にするデータ管理方法、データ検索方法、コンピュータ読み取り可能な記録媒体、データ管理装置及びデータ検索装置を提供することにある。

【 0 0 1 5 】

【課題を解決するための手段】

上記目的を達成するために、本発明のデータ管理方法は、管理すべき複数のデータを複数のデータ格納領域（セル）に別々に格納し、前記複数のデータ格納領域に格納されたデータ間の関連性に関する情報を前記複数のデータ格納領域あるいはこれら以外のデータ格納領域のそれぞれに付随した関連性情報可能領域（ふいさード）の少なくともいずれかに格納することを特徴とする。

【 0 0 1 6 】

ここで、前記関連性情報格納領域に格納される前記情報は、その関連性情報格納領域が付随したデータ格納領域とは異なる他のデータ格納領域を指定する情報を含むことを特徴とする。

【 0 0 1 7 】

また、前記他のデータ格納領域を指定する情報は、データ格納領域のポインタであることを特徴とする。

【 0 0 1 8 】

また、前記管理すべき複数のデータのいずれか2つ以上が同一である場合に、これら2つ以上の同一のデータをひとつのデータ格納領域に格納することを特徴とする。

【 0 0 1 9 】

また、前記同一のデータが格納されたデータ格納領域が、他のデータ格納領域に付随する関連性情報格納領域において2回以上指定される場合があることを特徴とする。

【0020】

また、検索パターンを識別するためのデータをデータ格納領域に格納し、このデータ格納領域に付随した関連性情報格納領域には、前記検索パターンのソーティング順序でソーティングされた前記情報を格納することを特徴とする。

【0021】

また、前記管理すべき複数のデータを更新した後に、前記他のデータ格納領域を指定する情報を更新されたデータに基づいて再びソーティングされた情報を前記関連性情報格納領域に格納することを特徴とする。

【0022】

また、前記ソーティング順序は、前記他の複数のデータ格納領域を指定する前記情報の配列順序をキー順位としたものであることを特徴とする。

【0023】

また、前記ソーティングされた情報の管理を、スプレイ・ツリー・アルゴリズムにより行うことを特徴とする。

【0024】

また、前記関連性情報格納領域に格納される前記情報は、その関連性情報格納領域が付随したデータ格納領域を指定する情報を含むことを特徴とする。

【0025】

また、複数の前記データ格納領域をひとつのデータ集団名（レキシコン）に対応させて管理することを特徴とする。

【0026】

また、前記管理すべき複数のデータの関連を表すために、前記データ格納領域を指定する情報の集合体である指定情報配列（コンテキスト）を形成し、前記指定情報配列の少なくとも一部を前記関連性情報格納領域に格納することを特徴とする。

【0027】

また、前記関連性情報格納領域に格納された情報から前記指定情報配列を辿ることが可能なことを特徴とする。

【0028】

一方、本発明のデータ検索方法は、管理すべき複数のデータが複数のデータ格納領域に別々に格納され、前記複数のデータ格納領域以外のデータ格納領域に付随した関連性情報可能領域に前記複数のデータ格納領域のそれぞれを指定する情報が所定の順序で格納されてなるデータベースから所定のデータを検索する検索方法であって、

前記関連性情報格納領域のうちで、検索条件と合致した順序で前記情報が格納されている関連性情報格納領域を検索するステップと、前記検索された前記関連性情報格納領域に格納されている前記情報のうちで、検索条件に合致した前記情報を検索するステップと、前記情報が指定する前記データ格納領域に格納されたデータを読み出すステップと、を備えたことを特徴とする。

【 0 0 2 9 】

一方、本発明の記録媒体は、データを管理するプログラムを格納したコンピュータ読み取り可能な記録媒体であって、前記プログラムは、

管理すべき複数のデータを前記コンピュータに付随した記憶装置に設けた複数のデータ格納領域に別々に格納させるステップと、前記複数のデータ格納領域に格納させたデータ間の関連性に関する情報を前記複数のデータ格納領域あるいはこれら以外のデータ格納領域のそれぞれに付随して設けた関連性情報格納領域の少なくともいずれかに格納させるステップと、を備えたことを特徴とする。

【 0 0 3 0 】

または、本発明の記録媒体は、データを検索するプログラムを格納したコンピュータ読み取り可能な記録媒体であって、前記プログラムは、

管理すべき複数のデータが複数のデータ格納領域に別々に格納され、前記複数のデータ格納領域以外のデータ格納領域に付随した関連性情報可能領域に前記複数のデータ格納領域のそれぞれを指定する情報が所定の順序で格納されてなるデータベースにおける前記関連性情報格納領域のうちで、検索条件と合致した順序で前記情報が格納されている関連性情報格納領域を検索するステップと、前記検索された前記関連性情報格納領域に格納されている前記情報のうちで、検索条件に合致した前記情報を検索するステップと、前記情報が指定する前記データ格納領域に格納されたデータを読み出すステップと、を備えたことを特徴とする。

【 0 0 3 1 】

一方、本発明のデータ管理装置は、データ入力部と、データ処理制御部と、データ記憶部と、を備えたデータ管理装置であって、

前記データ入力部は、管理すべき複数のデータを入力し、前記データ処理制御部は、前記管理すべき複数のデータを前記データ記憶部に設けた複数のデータ格納領域に別々に格納し、前記複数のデータ格納領域に格納したデータ間の関連性に関する情報を前記複数のデータ格納領域あるいはこれら以外のデータ格納領域のそれぞれに付随して前記データ記憶部に設けた関連性情報格納領域の少なくともいずれかに格納する、ことを特徴とする。

【 0 0 3 2 】

または、本発明のデータ管理装置は、データ入力部と、データ処理制御部と、データ記憶部と、を備えたデータ検索装置であって、

前記データ入力部は、検索条件を入力し、前記データ記憶部には、管理すべき複数のデータが複数のデータ格納領域に別々に格納され、前記複数のデータ格納領域以外のデータ格納領域に付随した関連性情報可能領域に前記複数のデータ格納領域のそれぞれを指定する情報が所定の順序で格納され、前記データ処理制御部は、前記データ記憶部が有する前記関連性情報格納領域のうちで、検索条件と合致した順序で前記情報が格納されている関連性情報格納領域を検索するステップと、前記検索された前記関連性情報格納領域に格納されている前記情報のうちで、検索条件に合致した前記情報を検索するステップと、前記情報が指定する前記データ格納領域に格納されたデータを読み出すステップと、を実行する、ことを特徴とする。

【 0 0 3 3 】

さらに、これらのデータ管理方法あるいはデータ検索方法を実行可能な各種の形態のソフトウェアも、本発明に含まれる。さらに、これらのデータ管理方法を応用した各種のアプリケーションも、本発明に含まれるものである。

【 0 0 3 4 】

ここで、「記録媒体」とは、例えば、ハードディスク（HD）、DVD-RAM、DVD-ROM、フレキシブル・ディスク（FD）やCD-ROMなどの他

に、RAMやROMなどの各種メモリも含む。

【 0 0 3 5 】

また、これらの媒体に記録されるべきプログラムを暗号化したり、変調をかけたり、圧縮したような状態で、イントラネットやインターネットなどの有線回線や無線回線を介してあるいは記録媒体に格納して頒布しても良い。

【 0 0 3 6 】

以上説明したように、本発明は、主記憶上でデータを効率良く管理する方式を提供することにより、上記問題を解決する。より具体的には、データの実体を格納するための領域（「データ格納領域」または「セル（cell）」と称する）に、セルに格納されたデータに関連した複数のセルの間の関係をポインタの列で表すデータ（「ロウ（row）」と称する）を格納する領域（「関連性情報格納領域」または「ファサード（facade）」と称する）を付随させたデータ構造とすることによって、データの効率的な操作（登録、更新、削除、検索）を実現するものである。

【 0 0 3 7 】

本発明によれば、このような構成を採用することによって、スキーマレスにデータの実体そのものを管理する事ができるようになり、また主記憶の特性（ポインタを効率良く扱うなど）を最大限利用したデータ管理アルゴリズムによって、RDBなどと比較して格段の性能を持つデータ管理システムが実現できるようになる。

【 0 0 3 8 】

また、本発明によれば、このような構成を採用することによって、データの効率的な操作が実現でき、しかもスキーマレスでデータが管理できるため、スキーマの枠による様々な制約から解放され、データ管理システムを用いたアプリケーションソフトウェアの開発生産性を著しく向上させることができるようになる。

【 0 0 3 9 】

【発明の実施の形態】

以下、図面を参照しつつ本発明の実施の形態について説明する。

本発明は、データの実体を格納するための領域（セル）に、セルに格納された

データに関連した複数のセルの間の関係をポインタの列で表すデータ（ロウ）を格納する領域（ファサード）を付随させたデータ構造とすることによって、データの効率的な操作（登録、更新、削除、検索）を実現するものである。このデータ構造を、「ファサード・ロウ構造」と称することにする。

個々のデータを格納するための基本データを「セル」と呼ぶ。

【0040】

図1は、セルの構造を表す概念図である。セルは、同図に表したように、C、C++、Javaなどのプログラム言語における1次元配列に相当する。セルは、格納するデータの「データ型」と「配列の長さ」（これを「セルサイズ」と称する）を指定して用いる。

【0041】

データ型には、byteストリームデータ、符号なし整数値（ビット長：8／16／32／64）、符号付き整数値（ビット長：8／16／32／64）、浮動小数点実数値、倍精度浮動小数点実数値、などがある。以下の説明では、図1に例示したように、セルの図表現を矩形により表し、セルに格納するデータを矩形の中に記すことにする。

【0042】

本発明におけるセルには、プログラム言語とは異なり、変数名に相当するものがない。その代わりに「ua (uni-space address)」と称するアドレスで指定できる。セルは配列なので、配列の位置を指定することにより、その内容にアクセスできる。その配列の位置を指定する値を「インデックス」と称する。インデックスは、図1に表したように0オリジン、すなわち0（ゼロ）から開始することができる。

【0043】

図2は、本発明における「セル」と「ファサード」との関係を表す概念図である。すなわち、セル (cell) が作成されると、同時にファサード (facade) が作成され、ファサードとセルは、同じuaによりアクセスできる。そして、ファサードに付随したセルに格納したデータを、ファサードの識別データと称する。従って、セルは、1つはデータの格納庫として、もう1つはファサードの識別子と

して使われる。すなわち、ファサード・ロウ構造では、データの内容（セル）とデータの管理（ファサード）とが一つの構造で扱われていて、RDBのようにデータ管理のための特別なデータ（テーブル）のようなものがないのが特徴である。

【0044】

セルとファサードの図表現は、図2のように、セルとファサードそれぞれを上下に連続した2つの矩形により表現することができる。上部がセルで、下部がファサードである。なお、セルにはファサードが必ず付随するが、ファサードには必ずしもロウが登録されているとは限らない。すなわち、空のファサードもありうる。ファサードにロウが登録されているかどうかを区別するため、図面においては、空のファサードと、ロウが登録されているファサードとを適宜区別して表す。

【0045】

セルは「レキシコン (lexicon)」により管理される。すなわち、「レキシコン」とは、「データ集団名」に相当する概念である。

【0046】

図3は、レキシコンにより管理されたセル群を例示する概念図である。同図に例示したように、セルは、必ずどこかのレキシコンに登録される。レキシコンは、複数存在し、それらは「レキシコン・セット (lexicon-set)」により管理される。それぞれのレキシコンには固有の名前が与えられており、この名前によってレキシコン・セットから特定のレキシコンを検索できる。アプリケーションはその処理に必要なレキシコンをその名前によってレキシコン・セットから検索し、次に、そのレキシコンの下に管理されているセルをその内容によって検索することにより、欲しいセルを得ることができる。

【0047】

図3は、レキシコン・セットから“従業員”レキシコンを検索し、次に、“斎藤”セルを検索するプロセスを例示している。

【0048】

セルは、データを格納する役割を有するが、格納するデータはそのセルが所属

するレキシコンの下には基本的には1つしか存在しないものとして行うことができる。つまり、同じ内容を持つ別のセルは存在せず、重複を許さないことを基本とすることができる。この場合には、セルはその内容と1対1に対応しており、データの実体を管理する機能として存在する。

【0049】

これに対して、セルの変数的な使い方もできる。この場合には、同じ内容のデータを異なるセルに格納することも可能である。これは、アプリケーションでどのようなデータを扱いたいかで決めれば良い。例えば、人名には「同姓同名」が存在する（つまり同姓同名で別人がある）が、それらを別物として管理したければ、同じデータではあるが別のセルに格納すれば良い。

【0050】

セルがレキシコンに登録されただけでは、単にデータがバラバラに存在するだけであり、セルの内容によりレキシコンから検索できるだけである。データが意味を持つためには、これらのセルが相互に関連を持ち、その関連がなんらかの方法で検索できなくてはならない。そのためには、セル間の関連を扱うための別な仕組みが必要になる。次に、このセル間の関連を操作する仕組みについて説明する。

【0051】

図4は、ある従業員データを構成する個々の要素をセルとし、“従業員”レキシコンに登録した状態を例示する概念図である。この状態では、セルがバラバラにレキシコンに連なっているだけで、セル同士の関連を扱うことはできない。しかし、アプリケーションでは、セルを「集合」として、すなわちセル間の関連をつけた1つのデータとして扱いたい。セル間の関連をつけた1つのデータとしては、例えば、表1を挙げることができる。

【0052】

【表 1】

表 1 設計データ

姓	入社年度	所属部門	役職	内線
佐藤	1981	本社	課長	6354
藤原	1981	研究所	主任	2406
小林	1980	研究所	部長	2409
田中	1974	本社	社長	6090
鈴木	1980	工場	課長	4390
斎藤	1981	工場	課長	3691

すなわち、この例においては、姓、入社年度、所属部門、役職、内線により区分された従業員の情報が与えられている。

【0053】

本発明によれば、このような管理すべきデータが、図4に例示したようにバラバラにセルに格納される。ここで、本発明の特徴のひとつは、重複したデータをひとつのセルで管理できる点にある。例えば、表1において、「入社年度」を見ると、6名の従業員のうちの3名の入社年度は「1981」である。このような場合に、本発明においては、「1981」に対応する3つのデータを格納する必要がなく、図4に例示したように、「1981」が格納されたセルをひとつ管理すれば良い。後に詳述するように、必要に応じて、このセルを何回でも指定できるからである。これは、「所属部門」や「役職」に関する重複データについても同様である。本発明によれば、このように重複するデータがある場合に、データ格納領域を大幅に節約できるという効果が得られる。

【0054】

次に、図4に表したように、「従業員」レキシコンに登録されているバラバラのデータを、表1の1行のデータ列、例えば「斎藤・1981・（工場）・課長・3691」のような1つの意味のあるデータ群として扱う方法について説明する

【 0 0 5 5 】

本発明においては、このようなデータの関連を扱うために、「コンテキスト (context)」と称するデータを用いる。本願明細書においては、「コンテキスト」を「指定情報配列」とも称する。

【 0 0 5 6 】

図5は、コンテキストを説明するための概念図である。すなわち、「コンテキスト (context)」は、セルとして登録されたデータの関連を扱うためのデータ管理方法において用いるデータであり、「リンク (link) 型」と称されるポインタデータの1次元配列である。リンク型のデータは、セルを指し示すデータ管理方法に固有の論理的なポインタデータである。「論理的なポインタ」という意味は、コンテキストに格納されているデータは、u aではなく管理用の情報を持ったデータである、ということである。つまり、コンテキストにはデータの実体は格納されず、データ実体を管理するセルを指し示すポインタにより構成される。コンテキストは、関連させたい複数のセルを指定するように形成される。そして、コンテキストに格納されているリンクデータから、セルを参照することができる。

【 0 0 5 7 】

本願において、リンクを図表現する際は、セル同様矩形で表し、リンクの指し示すセルを□で囲ってポインタ表現とする。例えば、リンクデータ[斎藤]は、斎藤という内容を持つセルを参照するリンクを表す。

【 0 0 5 8 】

コンテキストは、セルに格納された要素データ間の関係を表すが、データの関連性は、アプリケーションが必要とする検索方式に依存する。従って、コンテキストは、アプリケーションにおける設計データを基に構成することになる。

【 0 0 5 9 】

図6は、設計データとコンテキストとの関係を表す概念図である。

【 0 0 6 0 】

また、図7は、コンテキストを構成する処理手順を表すフローチャートである

【 0 0 6 1 】

図 6 に表した例では、図の上段の表の設計データを 2 種類の検索パターンで検索するアプリケーションを開発したいとする。これをデータ管理装置で実装するには、まず 2 種類の検索パターンを表現するデータを設計データに追加する必要がある。このデータは、後で説明するように、「ファサード」として機能することになる。このために、まず、図 7 にステップ S 1 及び S 2 として表したように、検索パターンのそれぞれにファサード名を割り当てる。例えば、2 つの検索パターンにそれぞれ“分類 1”と、“分類 2”という名前を付ける。

【 0 0 6 2 】

次に、ステップ S 3 において、元の設計データ（オリジナルデータ）の 1 行のデータを取り出す。そして、ステップ S 4 において、検索パターンのファサード名を各行のデータ列に追加する。さらに、図 4 に表したように、これら検索パターンのデータ（「分類 1」と「分類 2」）をレキシコンにも追加しておく。この段階では、1 行のデータは実体のデータ列である。この実体のデータ列を、実体を管理するセルを指し示すリンク列に置き換えるとコンテキストができあがる。これらの処理はステップ S 5 により、検索パターンの数だけ繰り返し、さらにステップ S 6 により、設計データの行数だけ繰り返す。

【 0 0 6 3 】

この例では、図 6 の下段に表したように、設計データに対応した 6 種類のコンテキストができる。

【 0 0 6 4 】

この例で見ても分かるように、コンテキストは複数のセルの間の関連を表現しているだけであり、コンテキストを直接検索する方法はない。そこで、コンテキストで間接的に表わされたデータをハンドリングするための新たな機構が必要になる。次にその機構について説明する。

【 0 0 6 5 】

コンテキストを検索するには、これを「ロウ」としてファサードに登録する必要がある。登録するファサードは、設計データに追加した検索パターンを表すデ

ータに付随したファサードである。

【0066】

図8は、ファサードに登録されたロウの一例を表す概念図である。この例では、コンテキストの中に格納されている検索パターンを表すリンクが示す、“分類1”のファサードに登録している。コンテキストをロウとしてファサードに登録する時は、コンテキストを構成するどのデータ項目を検索の「キー」にするかをキーの順位とともに指定する。図8の例では、第1キーとして入社年度、第2キーとして所属部門が指定されている。また、この際に、ファサードを表すデータ（図8の例では「分類1」）を第0キーとして必ず指定しなければならない。ここで、ファサード内でのロウの順位を「スロット (slot)」と称することにする。

【0067】

ロウは、登録する際に指定されたキー順にコンテキストのリンクデータを並び替えられて作られる。また、コンテキストをロウとしてファサードに登録する際には、コンテキストのすべてのデータを登録する必要はない。例えば、図8の例では“分類1”ファサードに登録したロウには、他の検索で使用する“分類2”は使う必要が無いので、“分類2”はロウには登録していない。ロウに格納されているデータは、コンテキストを構成しているリンクデータを並び替えたものである。例えば、図8の例では、第1キーに[1981]、第2キーに[本社]が指定されて“分類1”ファサードに登録され、このコンテキストに対応するロウは第4スロットに位置している。

【0068】

以上の説明から分かるように、ロウはコンテキストを元にしてファサードに登録されるが、その内容はロウとして登録する際に指定したキー順位で以下の如く並びかえられたものである。

コンテキスト：[分類1] [分類2] [佐藤] [1981] [本社] [課長] [6354]

↓

ロウ：[分類1] [1981] [本社] [佐藤] [課長] [6354]

【0069】

なお、コンテキストもロウも内部ではセルとして扱われているので、それぞれ個別に *u a* を持っており、セル同様に *u a* でアクセスすることができる。そして、データ管理装置による検索などの処理結果の戻り値は *u a* で返ってくる。

【0070】

以上で説明したデータ管理方式の実現方法であるが、例えばこの構造を、スプレイ・ツリー (splay-tree) アルゴリズムで実装することが考えられる。なお、スプレイ・ツリー・アルゴリズムについて開示した文献としては、例えば、Robert E. Tarjan, "Data Structure and Network Algorithms", the Society for Industrial and Applied Mathematics, 1983 を挙げるができる。スプレイ・ツリー・アルゴリズムは、動的に変更する2分木で、データアクセスが発生する毎にスプレイング (splaying) 操作を実行する。

【0071】

図9は、スプレイ・ツリー・アルゴリズムにおけるスプレイング操作を例示する概念図である。図9において、スプレイング操作は、次のようなステップで行う。例えば、検索処理を行うとルートからノードを検索キーと比較しながら降りてきて、最終的に検索対象に到達する。そのノードを (x) とすると、(x) の前後のツリーの構造が図9に表した3種類のパターンのいずれかに該当するので、それを選んでツリーの構造を変える。この処理で、(x) はツリーの構造を1段上がることになるが、今度は新しい位置で、再度 (x) の前後のツリー構造を見て図9のいずれかの構造を選び、同様に処理する。この処理を続けると、最終的にはツリーのルート (頂点) に到達する。

【0072】

このようなスプレイング操作を通じて、偏りのあるデータアクセスに対して最適な構成を取る2分木ができる。これをファサードにおけるロウの管理に用いることでデータをソートして管理することができる。

【0073】

スプレイ・ツリー・アルゴリズム以外にもデータの管理法は各種知られている

が、これらのデータ管理方式を用いても同様の効果を得ることができる。本発明においては、図 1 乃至図 8 に関して前述したデータ管理構造の方式が本質的な部分であり、物理的な実装方式には依らずに上記効果を発揮することができる。

【0074】

【実施例】

以下、具体的な実施例を参照しつつ本発明の実施の形態についてさらに詳細に説明する。本実施例においては、表 1 に表した設計データを用い、本発明のデータ管理方法をどのように実装するかを説明する。

【0075】

まず、本実施例における問題設定を説明する。扱いたいデータは、表 2 に表した従業員データである。

【0076】

【表 2】

表 2 設計データ

姓	入社年度	所属部門	役職	内線
佐藤	1981	本社	課長	6354
藤原	1981	研究所	主任	2406
小林	1980	研究所	部長	2409
田中	1974	本社	社長	6090
鈴木	1980	工場	課長	4390
斎藤	1981	工場	課長	3691

この表から分かるように、データ項目は、「姓」、「入社年度」、「所属部門」、「役職」、「内線」の 5 種類であり、全部で 6 件のデータからなる。このようなデータを 2 種類の検索パターンで分類するとしよう。第 1 の検索パターンは、入社年度を第 1 キーに、所属部門を第 2 キーにするもので、第 2 の検索パター

ンは内線を第1キーに、姓を第2キーにするものであるとする。

【0077】

以下、コンテキストからどのようにロウが形成され、どのようにファサードに登録されるかを説明する。

【0078】

図10は、本実施例におけるデータ管理方法の手順を表すフローチャートである。

【0079】

まず最初に、ステップS11において、設計データを構成するセルをレキシコンに登録する。図11は、セルがレキシコンに登録された状態を表す概念図である。図示した如く、レキシコンとセルのツリーができあがる。

【0080】

次に、ステップS12においてコンテキストを形成する。本実施例におけるコンテキストは、例えば、図11の右下に表した如くである。

【0081】

次に、ステップS13において、これらのコンテキストからロウを形成し、これを所定のファサードに登録する。本実施例においては、コンテキストは、“分類1”と“分類2”のそれぞれのセルに付随したファサードに設計データに表された検索パターンで登録される。

【0082】

具体的には、まず最初に、図12に表したように、“分類1”ファサードに、同図に表したキー順位で登録する。同図に表した検索パターンが、“分類1”に対応した検索パターンである。

【0083】

その結果、図13に表したように、“分類1”ファサードには1件のロウが登録される。すなわち、コンテキストは、指定された検索キーの順番に従い、以下のように、リンクデータの配列としてのロウに変換される。

コンテキスト：[分類1][分類2][佐藤][1981][本社][課長][6354]

↓

ロウ : [分類1] [1981] [本社] [佐藤] [課長] [6354]

【 0 0 8 4 】

なお、ここで、[分類2] がロウには登録されないこと、及び、[分類1] が第0カラム（第0キー）として登録されることに注意されたい。この登録の結果として、“分類1” ファサードには1件のロウが登録されるが、その他のセルに付随したファサードは相変わらず空のままである。

【 0 0 8 5 】

次に、ステップ S 1 4 において検索パターンの数だけ繰り返す。

【 0 0 8 6 】

具体的には、図 1 4 に表したように、“分類2” ファサードにロウを登録する。この際には、同図に表したキー順位で登録する。その結果、“分類2” ファサードには1件のロウが図 1 5 のように登録される。すなわち、コンテキストは、指定された検索キーの順番に従い、以下のように、リンクデータの配列としてのロウに変換される。

コンテキスト : [分類1] [分類2] [佐藤] [1981] [本社] [課長] [6354]

↓

ロウ : [分類2] [6354] [佐藤] [1981] [本社] [課長]

【 0 0 8 7 】

ここでも、“分類1” ファサードへの登録と同様に、[分類1] がロウには登録されないこと、及び、[分類2] が第0カラム（第0キー）として登録されることに注意されたい。この登録の結果として、“分類1” ファサードと“分類2” ファサードにはそれぞれ1件のロウが登録されるが、その他のセルに付随するファサードの空のままである。

【 0 0 8 8 】

以下、ステップ S 1 5 において、コンテキストの件数だけ繰り返す。

【 0 0 8 9 】

まず、2番目のコンテキストを、前と同様に”分類1”ファサードと”分類2”ファサードに登録する。図16は、このようにしてそれぞれのファサードに2件のロウが登録された状態を表す概念図である。ロウはそれぞれのファサードで検索キー順にソートされる。

【 0 0 9 0 】

以下、このようにして6件すべてのコンテキストをロウに変換してファサードに登録する。図17は、6件のロウが登録された状態を表す概念図である。これですべてのデータが登録され、検索の準備ができたことになる。

【 0 0 9 1 】

次に、本発明のデータ管理方法におけるファサード・ロウ構造を用いた検索手順について説明する。

【 0 0 9 2 】

図18は、本検索手順を表すフローチャートである。

【 0 0 9 3 】

また、図19は、本検索手順を説明するための概念図である。

【 0 0 9 4 】

本発明においては、検索のために、まずステップS21において、検索したいパターンに対応するファサードを検索する。具体的には、検索のトランザクション関数が起動されると、まず最初に、与えられたレキシコン・セットから、トランザクション関数の処理に必要となるレキシコンをその名前により検索する。例えば、本実施例の場合は、”従業員”レキシコンを検索する。

【 0 0 9 5 】

次に、トランザクション関数で必要とするデータをそのレキシコンにおいて検索する。最初にファサードを探してこなくてはならないが、これはファサードと対応しているセルをレキシコンから探せば良い。本実施例では、”分類1”セルをレキシコンから検索してくる。

【 0 0 9 6 】

次に、ステップS22において、ファサード内に登録されているロウを所望の

キーで検索する。本実施例の場合は、第 1 キー＝” 1 9 8 1 ”、第 2 キー＝” 本社 ” により検索して所望のロウを獲得する。

【 0 0 9 7 】

次に、ステップ S 2 3 において、ロウの中の所望のセルを、リンクを辿ることにより求める。すなわち、得られたロウの中からトランザクション関数に必要なデータを見つけ出す。例えば、本実施例においては、検索されたデータの「姓」の項が欲しいので、ロウ内の 3 番目（0 オリジンなので）のデータ項目であるリンクが指し示すセルから、所望のデータである” 佐藤 ” を得る。

【 0 0 9 8 】

以上説明した具体例の場合は、ロウから直接セルを参照しているが、もっと複雑な検索の場合は、コンテキストに溯って別なデータへアクセスしたりすることも可能である。いずれにしても、基本となる検索手順はファサードからロウを確定することであり、これを中心に検索処理を組み立てればよい。

【 0 0 9 9 】

次に、本発明のデータ管理方法を実行可能なデータ管理装置について説明する。

【 0 1 0 0 】

図 2 0 は、上述したデータ管理方法をハードウェア上に実現したデータ管理装置の概観を例示する鳥瞰図である。

【 0 1 0 1 】

また、図 2 1 は、その要部構成を例示したブロック図である。

【 0 1 0 2 】

このデータ管理装置 8 0 の本体は、データ入力部 8 0 A と、データ処理制御部 8 0 B と、データ記憶部 8 0 C と、データ出力部 8 0 D と、を有する。

【 0 1 0 3 】

データ入力部 8 0 A は、管理すべきデータを外部から入力する役割を有し、その入力手段としては、例えば、フロッピーディスク装置（フロッピーディスクドライブ） 8 1、光ディスク装置（光ディスクドライブ） 8 2 などを具備している。フロッピーディスクドライブ 8 1 に対してはフロッピーディスク 8 3 を、また

光ディスクドライブ 82 に対しては CD-ROM あるいは DVD ディスクなどの光ディスク 84 をその挿入口から挿入し、所定の読み出し操作を行うことにより、これらの記録媒体に格納されたデータをシステム内に入力することができる。また、所定のドライブ装置を接続することにより、例えば半導体メモリ装置としての ROM 85 や、磁気テープ装置としてのカセットテープ 86 を用いることもできる。さらに、キーボード 87 によりデータを入力したり、ネットワーク回線 88 を介して、他のコンピュータあるいはデータ出力装置からデータを入力することもできる。

【0104】

このようにして入力されたデータは、データ記憶部 80C に格納される。この際に、図 1 乃至図 19 に関して前述したように、レキシコン・セット、レキシコン、セル、ファサード、コンテキスト、ロウなどが適宜形成される。これら一連の処理は、データ処理制御部 80B により実行される。また、ロウのソーティングや、ファサードに格納されたデータのスプレイングも、データ処理制御部 80B により実行される。

【0105】

このようにして管理され、ソートされたデータは、必要に応じてデータ出力部 80D により出力される。データ出力部 80D は、例えば、ディスプレイ 89 や、フロッピーディスク 83 などの各種の媒体、あるいはネットワーク回線 88 を介して所望のデータを外部に出力する。

【0106】

後に詳述するように、本発明においては、データを格納するデータ記憶部 80C として、RAM などの半導体メモリを用いることができる。その結果として、高速にデータを展開、管理することが可能となり、データ管理の性能を従来よりも大幅に改善することができる。

【0107】

なお、本発明は、図 1 乃至図 19 に関して前述したようなデータ管理方法が実行可能なソフトウェアも包含する。このようなソフトウェアは、データ読み取り部 80A に関して前述したものと同様に、光ディスク 84 などの記録媒体に格納

して、管理装置 8 0 のデータ処理制御部 8 0 B にダウンロードすることが可能である。または、このようなソフトウェアは、ネットワーク回線 8 8 を介してダウンロードすることもできる。

【0108】

以上説明した本発明のデータ処理方法及び処理装置は以下に詳述する各種の効果を発揮する。

【0109】

(1) 超高速データ処理機能

本発明のデータ管理方法は、主記憶メモリ（以下、メモリと呼ぶ）上でデータを管理することが容易である。そして、本発明のデータ管理方式は、メモリの特性を活かした最適なデータ構造とアルゴリズムを実装することが容易である。このため、ハードディスク装置上に構築した従来の DBMS (database management system) と比べて格段の高性能を実現することができる。

【0110】

一方、RDB はハードディスク装置上でデータを管理することを前提としたアルゴリズムである。RDB においてデータキャッシュ機能を用いてメモリへ全データを載せて（フルキャッシュ状態）処理を行う場合と比較しても、本発明のデータ管理方法の方が 2 桁以上高速のデータ処理を実現できる。より詳しく説明すると、通常の PC（クロック周波数 4 5 0 M H z Pentium III（商標）

1 CPU 構成）サーバを用いた TPC-C ベンチマークで、本発明の管理方法は、1 6 0 0 T P S を実現する。この値は、現在（2 0 0 0 年 4 月）の TPC-C ベンチマークにおいて最高レベルの性能である。これは、最高速のハードウェアを用いて実現できる処理性能を PC で容易に実現できることを意味している。

【0111】

一方、本発明のデータ管理方法は、メモリ上で実現された DBMS なのでメモリの使用効率が問題となるが、効率の良いデータ管理アルゴリズムの採用しているため、RDB フルキャッシュと同程度のメモリ実装効率を実現できる。

【0112】

本発明のデータ管理方法の 1 つの特徴は、データの型式に依らない処理性能が

実現されることである。すなわち、本発明のデータ管理方法は、どのような型式のデータでも扱うことができ、データ型式に性能が影響されることが無いため、どのようなデータの型式に対しても、上述した最高性能が保証される。例えば、RDBのデータモデルである表データに加え、経路検索やCAD (computer aided design) などで必要となるネットワーク構造のデータや、画像や音声などのマルチメディアデータなど、RDBでは効率良い実装が難しいデータ構造でも性能劣化がない。

【 0 1 1 3 】

(2) 柔軟なシステム構築機能

本発明のデータ管理方法は、分散システムを構築するための様々な機能を持っている。すなわち、本発明のデータ管理方法において提供している分散システム構築機能は、リモート／ローカル構造と呼ばれるシステムのハードウェア構成に依存しない方式であるため、アプリケーション・ソフトウェアはシステムのハードウェア構成に関する実装を行う必要が無い。このため、ソフトウェア開発をシステム構築から独立させることができる。すなわち、システム構築のためのコーディングは一切不要で、システム構築／設定作業はすべてシステム運用環境を用いて行うことができる。

【 0 1 1 4 】

また、本発明のデータ管理方法は、スケーラビリティの高いシステムを実現することができる。高いスケーラビリティの実現は、1つにはデータサイズやトランザクション量の増加に対する性能の劣化の度合いが低いことによる。本発明のデータ管理方法の処理性能は、データのサイズを n とする時、 $O(\log(n))$ で実現できる(O :オーダー)ため、データサイズによる処理性能の劣化が非常に緩やかである。また、多くのDBMSでは性能のクリティカルポイントがあり、データサイズやトランザクション量がこれを超えると急激に性能が劣化するが、本発明のデータ管理方法では、そのようなことが無い。

【 0 1 1 5 】

もう1つの高スケーラビリティの実現方法は、システム構成の負荷分散機能による。負荷分散機能により、設計値を超えるデータサイズやトランザクションが

増えた場合には、コンピュータを新たに追加してトランザクションやデータを振り分けて処理を分散させる事によって対処することができる。

【 0 1 1 6 】

本発明のデータ管理方法においては、システム設定に関して、データ構造の変更やトランザクション関数の追加などの機能については動的に変更することができる。そのため、これらの機能についてはシステムを停止させることなく機能変更ができ、システム設定変更の対応が非常に楽である。

【 0 1 1 7 】

さらに、本発明のデータ管理方法は、プラットフォーム非依存のDBMSである。本発明のデータ管理方法が対応可能なOSとしては、Solaris、Windows/NT、Linux（将来機能）などが挙げられ、データ管理方法のI/Fやデータ構造はプラットフォーム非依存で作られているので、マルチプラットフォームのシステム構築が可能である。

【 0 1 1 8 】

また、本発明のデータ管理方法において、業務処理はトランザクション関数としてデータ管理装置に組み込んで用いるが、トランザクション関数はJavaで記述することができる（将来機能）ため、トランザクション関数のプラットフォームを超えた利用が可能になり、例えばトランザクション関数の配信などが実現できる。

【 0 1 1 9 】

これらの諸機能を組み合わせることによって、例えば従来のCPU構成よりも1桁以上小さい構成で従来と同等以上の性能が実現でき、ノート型PC、場合によってはパームトップPCを複数台つなげてシステムを組むことなど、従来ではできなかったシステム構成も可能となり、幅広いシステム構築の選択肢を得ることができる。

【 0 1 2 0 】

(3) 開発生産性向上機能

本発明のデータ管理方法は、以下の4つの特徴から、アプリケーション開発に高い生産性を与えることができる。

【0121】

第1の特徴は、本発明のデータ管理方法において用いるデータ構造による。本発明におけるデータ構造は、従来型のDBMSと異なり、スキーマは存在せず直接実体のデータを扱うものとなっている。そのため、スキーマの枠の制約を受けることなく、アプリケーション開発者の望む任意のデータ構造を実現することができる。従って、既存のスキーマに無理矢理にあわせるなどの余計な処理を実装することがなくなり、その結果として生産性や保守性が大幅に向上する。特に、ネットワーク型データやマルチメディアデータなどは、RDBで実装するとRDBのデータモデルである表データに無理矢理合わせなくてはならず、アプリケーション内部のデータ構造への非常に面倒な変換作業をアプリケーションで実装しなくてはならず、アプリケーションの開発効率が著しく低下する。このような複雑なデータでもデータ管理装置では素直に設計、実装することができ、RDBと比べるとアプリケーション開発生産性が高い。

【0122】

第2の特徴は、トランザクション関数にある。本発明のデータ管理方法においては、業務アプリケーションはトランザクション関数として実装され、データへのアクセスはトランザクション関数を通じて行われる。すなわち、トランザクション関数によってデータのカプセル化を実現している。データのカプセル化は、データ構造の変更がアプリケーションへ波及しないため、データ操作を行うトランザクションを部品化できる。また、データ管理装置はリモート/ローカル構造のシステム構成を取っているため、トランザクション関数はシステムに関する実装を一切含まない。そのため高い独立性を持つ部品ができ再利用性が一層高まる。トランザクション関数を再利用することで開発生産性が著しく向上する。

【0123】

第3の特徴は、リモート/ローカル構造のシステム構造にある。リモート/ローカル構造は、アプリケーション・ソフトウェアをコンピュータのハードウェア構成から独立させる。そのため、アプリケーションはシステム物理構造、すなわちハードウェアの物理構成やデータがどのコンピュータ上に実装されているかあるいはデータ整合性など、をまったく意識する必要がなく、業務ロジックの設計

・実装に専念でき、開発生産性が向上する。

【 0 1 2 4 】

第4の特徴は、高速性能による。本発明のデータ管理方法が提供する高速性能による恩恵は、アプリケーションの開発生産性にも寄与する。例えば、従来ならパラレル処理などの姑息な技法を用いなければ実現できないような性能を素直な設計で実現できるため、すっきりとしたソフトウェア構造が実現でき、そのためプログラムの理解性が優れ結果として保守性が向上し、さらには機能拡張も容易となる。さらに、データ規模拡大による影響が少ないので、余計な設計検討をする必要も無い。また、本発明におけるデータの処理方式は、データ構造に依らず性能が決定できるため、データ構造と処理内容が決まれば性能を簡単に予測することができ、従来のシステムでは難しかった性能設計が容易にできる。加えて、要求性能を実現するための性能チューニングなどの後戻り処理も発生しない。

【 0 1 2 5 】

(4) 高信頼化機能

本発明のデータ管理方法は、以下の4点の信頼性向上機能を持っている。

【 0 1 2 6 】

第1の機能は、信頼性強化機能である。本発明のデータ管理方法は、ネットワークを活用した多様な信頼性スペックが実現できる機能をもっている。1つにはバックアップ機能である。これはデータのハードディスク装置などへの保全をするための機能で、システムの初期化や障害発生時にバックアップファイルから再ロードすることができる。もう1つの機能として、システムの多重化機能とデータのリカバリ機能により、障害発生時の速やかなシステムリカバリ処理が可能である。第3には信頼性向上のツールと連携機能である。本発明のデータ管理方法は、例えば、複数のCPUを用いて、ハードウェア障害が発生した時、バックアップ側のシステムに引き継ぐことができるツールが製品化されているが、これと組み合わせることによってハードウェア障害に対処できるようになる。これらの諸機能はシステム管理環境を用いて簡単に設定可能で、ファイルへのバック・アップ・サイトの設定機能などを通じて実現できる。

【 0 1 2 7 】

第2の機能は、トランザクションのシリアル化である。分散環境でのトランザクション関数の実行においてはデータの整合性が問題になるが、データ管理装置ではトランザクション関数はシリアル化され実行されるのでデータの整合性が完全に保証される。

【0128】

第3の機能は、データのカプセル化である。データ管理装置で管理するデータはトランザクション関数のみアクセスでき、外部からはデータは隠蔽されているため、予期せぬデータのアクセスによるデータ破壊を防ぎ、データの信頼性が大幅に向上する。

【0129】

第4の機能は、リモート／ローカル構造のシステム構造にある。前述のように、リモート／ローカル構造によって、アプリケーション・ソフトウェアはコンピュータのハードウェア構成から独立になるため、アプリケーションはシステム物理構造に関する実装が不要になり、その結果として解決する事が非常に難しい物理的な原因によるバグやトラブルから開放される。

【0130】

【発明の効果】

以上詳述したように、本発明によれば、従来よりも低コストのハードウェアを用いつつ各種性能が大幅に向上したデータ管理を実現することが可能となり、産業上のメリットは多大である。

【図面の簡単な説明】

【図1】

本発明におけるセルの構造を表す概念図である。

【図2】

本発明における「セル」と「ファサード」との関係を表す概念図である。

【図3】

レキシコンにより管理されたセル群を例示する概念図である。

【図4】

ある従業員データを構成する個々の要素をセルとし、“従業員”レキシコンに登

録した状態を例示する概念図である。

【図 5】

コンテキストを説明するための概念図である。

【図 6】

設計データとコンテキストとの関係を表す概念図である。

【図 7】

コンテキストを構成する処理手順を表すフローチャートである。

【図 8】

ファサードに登録されたロウの一例を表す概念図である。

【図 9】

スプレイ・ツリー・アルゴリズムにおけるスプレイング操作を例示する概念図である。

【図 1 0】

本発明の実施例におけるデータ管理方法の手順を表すフローチャートである。

【図 1 1】

セルがレキシコンに登録された状態を表す概念図である。

【図 1 2】

” 分類 1 ” ファサードに、同図に表したキー順位で登録する様子を表す概念図である。

【図 1 3】

” 分類 1 ” ファサードに 1 件のロウが登録された様子を表す概念図である。

【図 1 4】

” 分類 2 ” ファサードにロウを登録する様子を表す概念図である。

【図 1 5】

” 分類 2 ” ファサードに 1 件のロウが登録された状態を表す概念図である。

【図 1 6】

それぞれのファサードに 2 件のロウが登録された状態を表す概念図である。

【図 1 7】

6 件のロウが登録された状態を表す概念図である。

【図 1 8】

本発明における検索手順を表すフローチャートである。

【図 1 9】

本発明における検索手順を説明するための概念図である。

【図 2 0】

本発明のデータ管理方法をハードウェア上に実現したデータ管理装置の概観を例示する鳥瞰図である。

【図 2 1】

本発明のデータ管理装置の要部構成を例示したブロック図である。

【符号の説明】

8 0 データ管理装置

8 0 A データ入力部

8 0 B データ処理制御部

8 0 C データ記憶部

8 0 D データ出力部

8 1 フロッピーディスク装置（フロッピーディスクドライブ）

8 2 光ディスク装置（光ディスクドライブ）

8 3 フロッピーディスク

8 4 光ディスク

8 5 ROM

8 6 カセットテープ

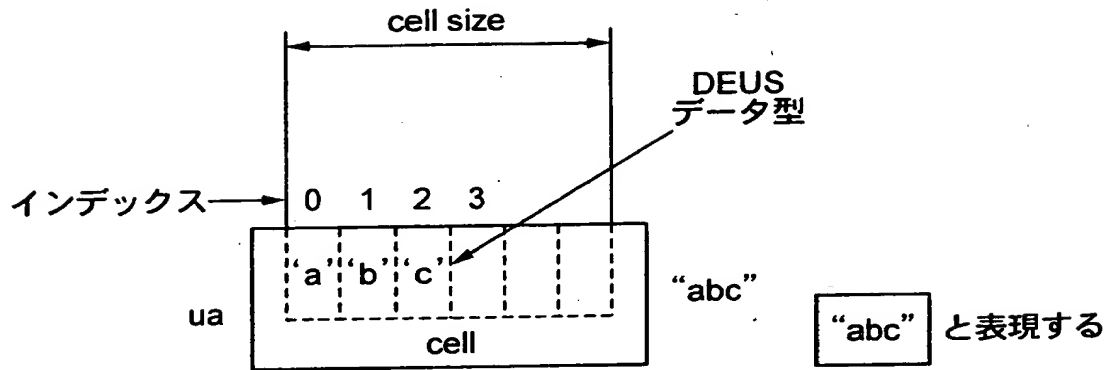
8 7 キーボード

8 8 ネットワーク回線

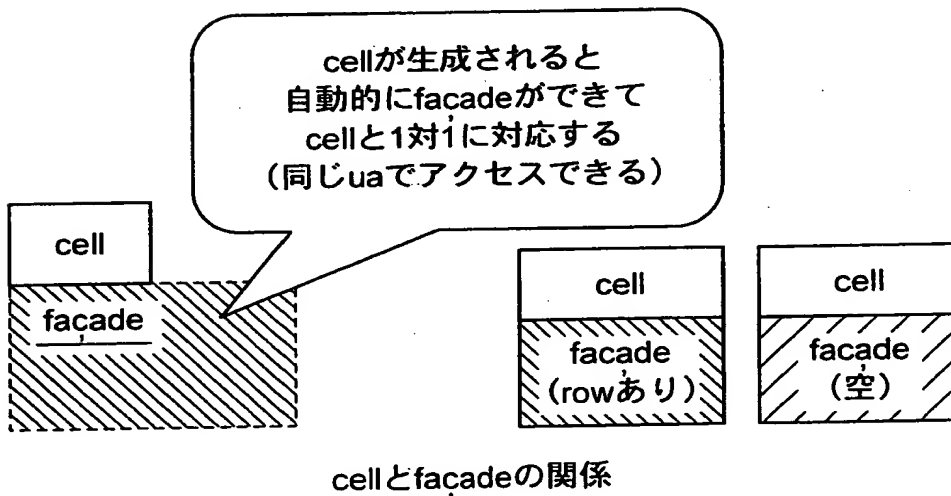
8 9 ディスプレイ

【書類名】 図面

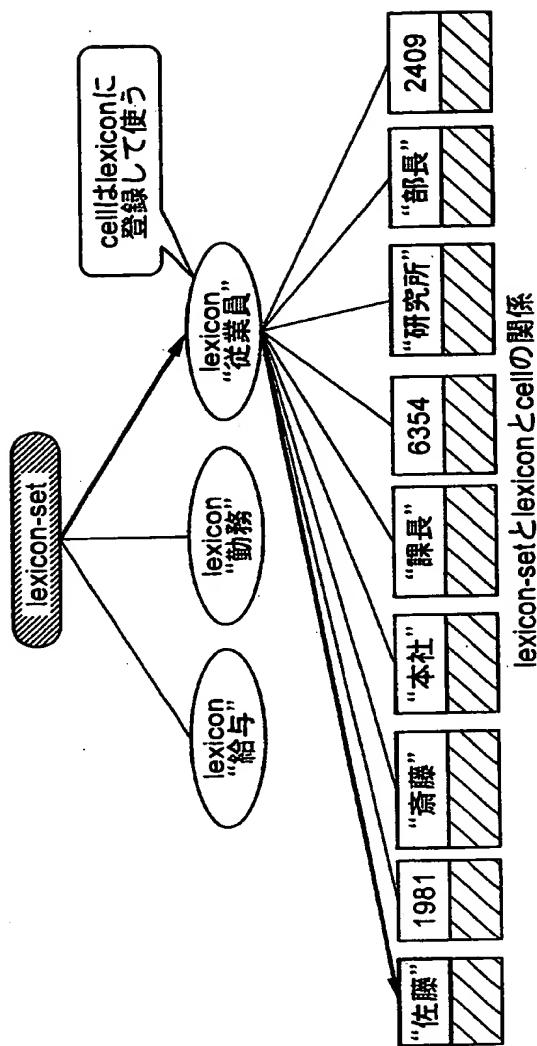
【図 1】



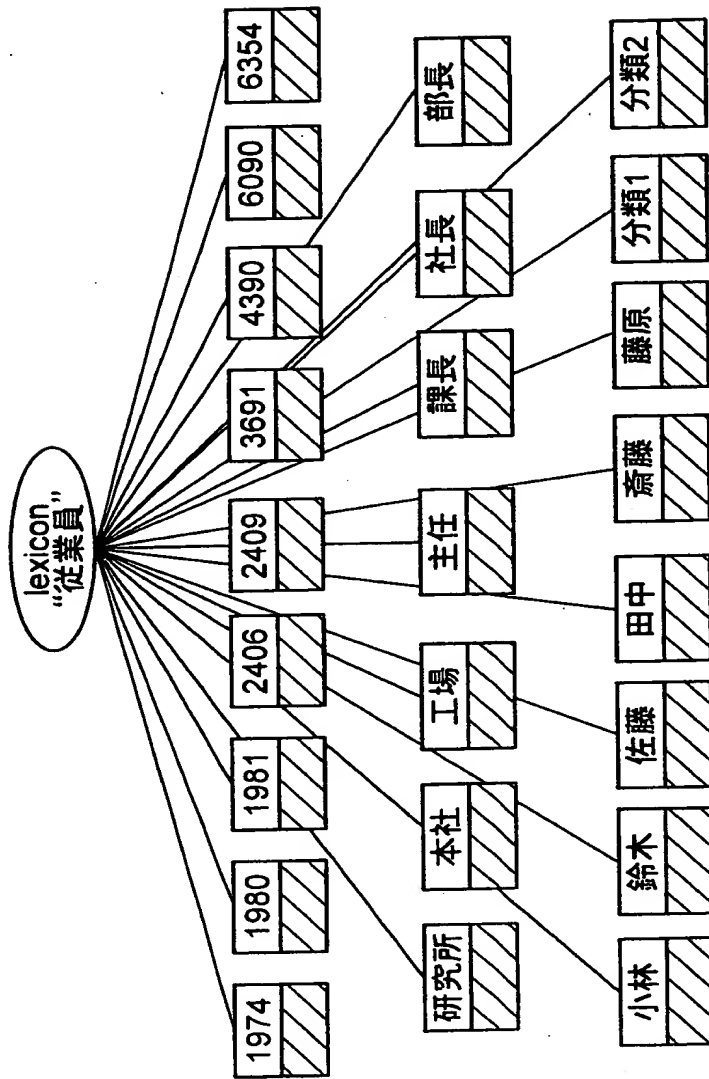
【図 2】



【図3】

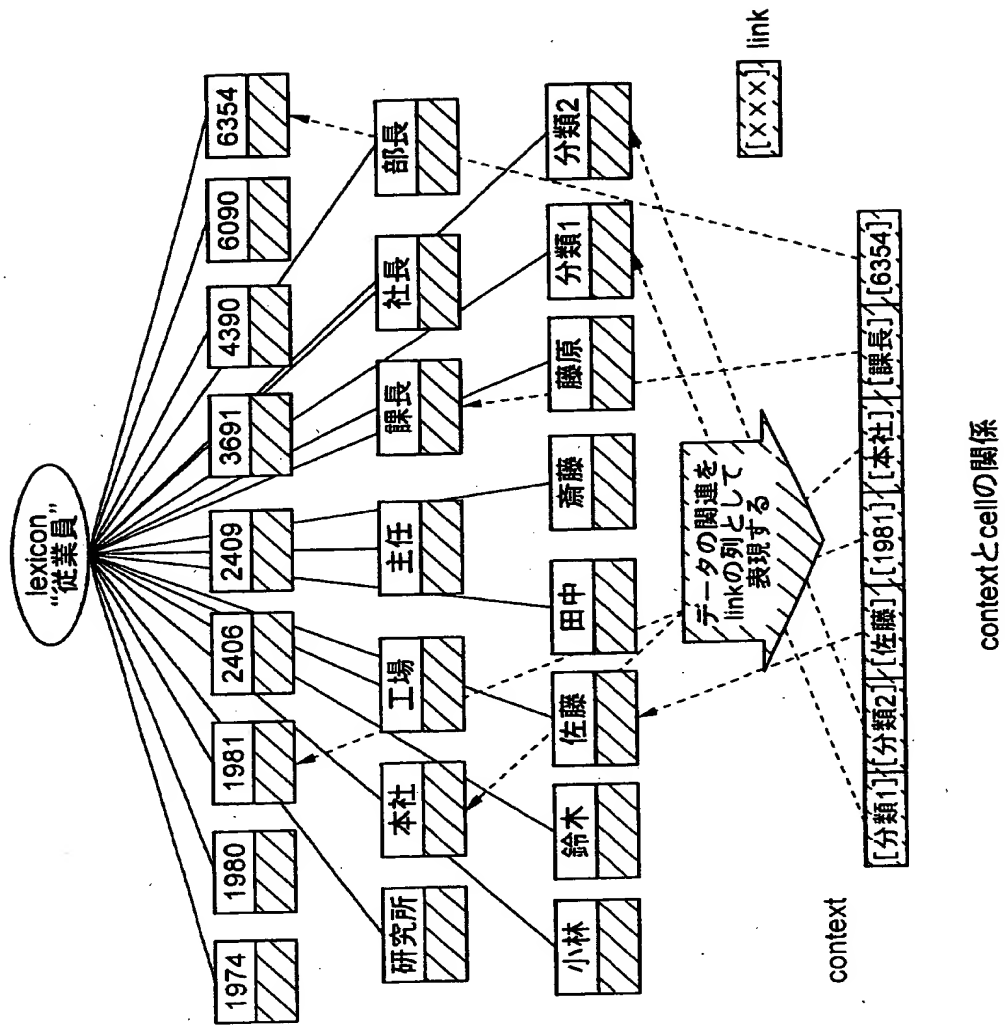


【図4】



lexiconにcellが登録された

【図 5】



【図 6】

姓	入社年度	所属部門	役職	内線
佐藤	1981	本社	課長	6354
藤原	1981	研究所	主任	2406
小林	1980	研究所	部長	2409
田中	1974	本社	社長	6090
鈴木	1980	工場	課長	4390
斎藤	1981	工場	課長	3691

設計データ

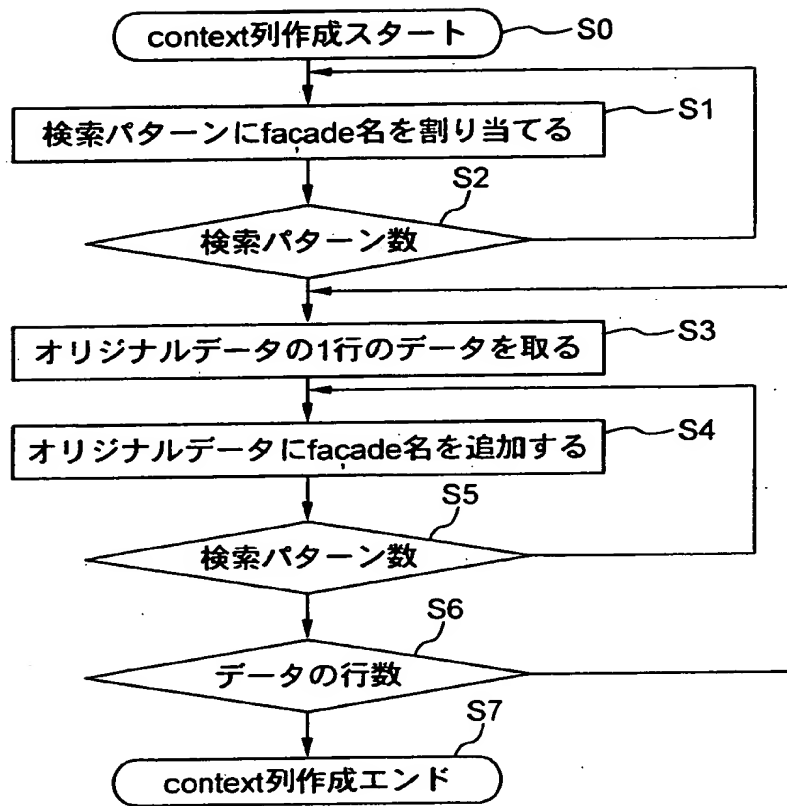


[分類1]	[分類2]	[佐藤]	[1981]	[本社]	[課長]	[6354]
[分類1]	[分類2]	[藤原]	[1981]	[研究所]	[主任]	[2406]
[分類1]	[分類2]	[小林]	[1980]	[研究所]	[部長]	[2409]
[分類1]	[分類2]	[田中]	[1974]	[本社]	[社長]	[6090]
[分類1]	[分類2]	[鈴木]	[1980]	[工場]	[課長]	[4390]
[分類1]	[分類2]	[斎藤]	[1981]	[工場]	[課長]	[3691]

contextの列

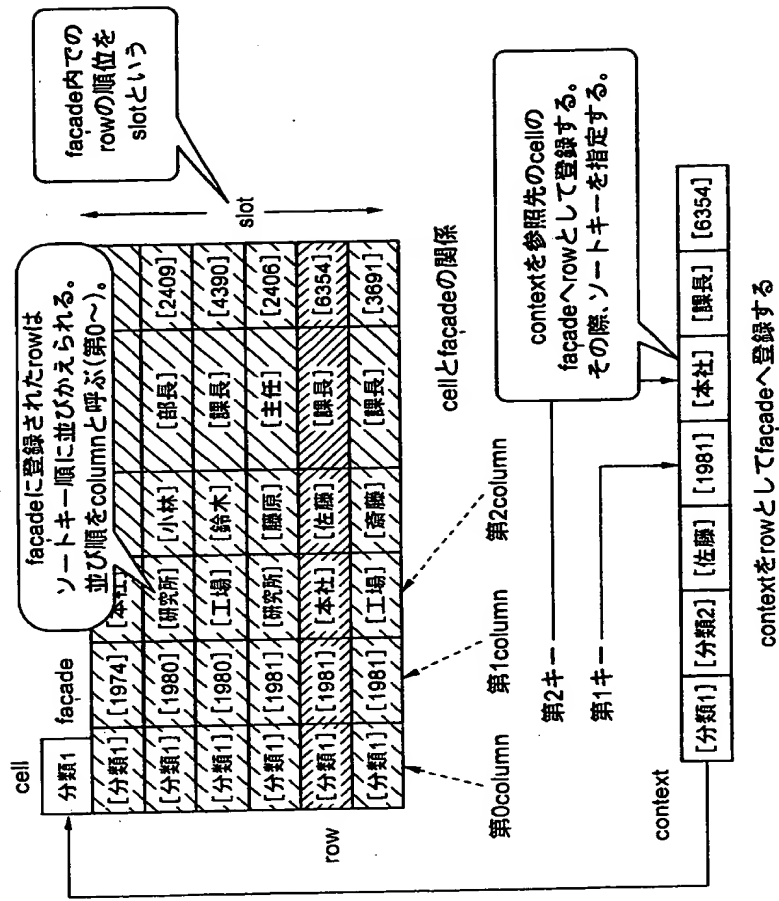
設計データからcontextの構成法

【図 7】

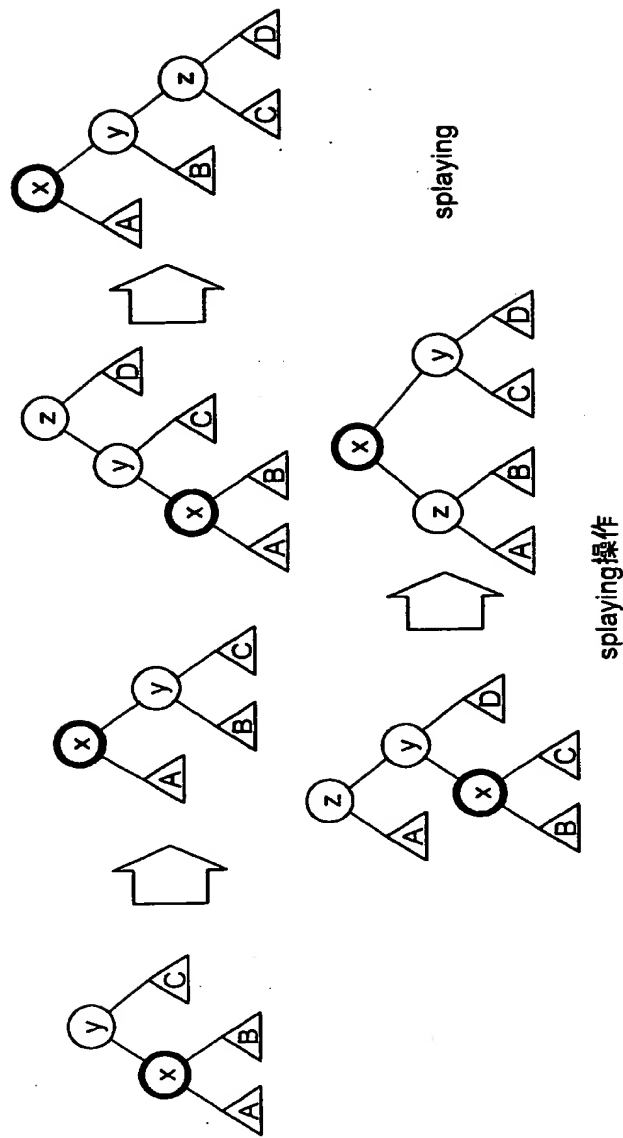


設計データからのcontextの構成法:フローチャート

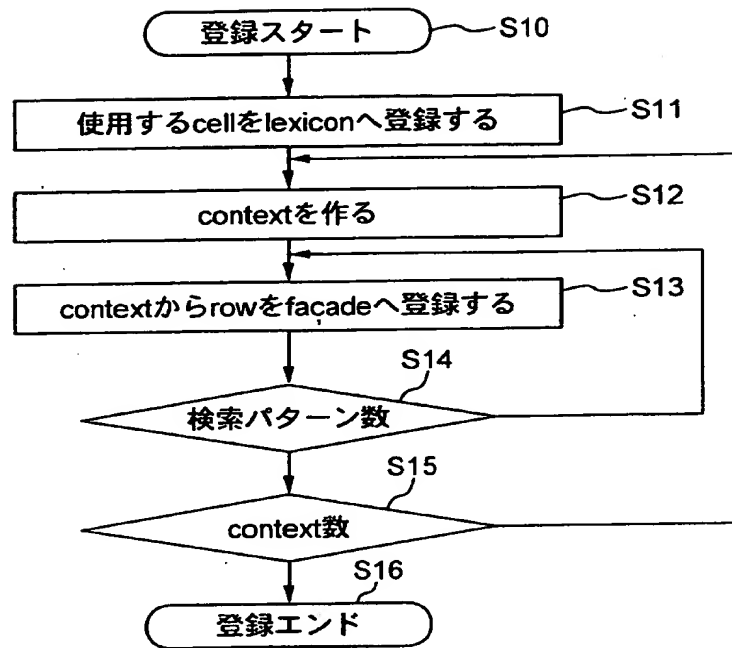
【図 8】



【図9】

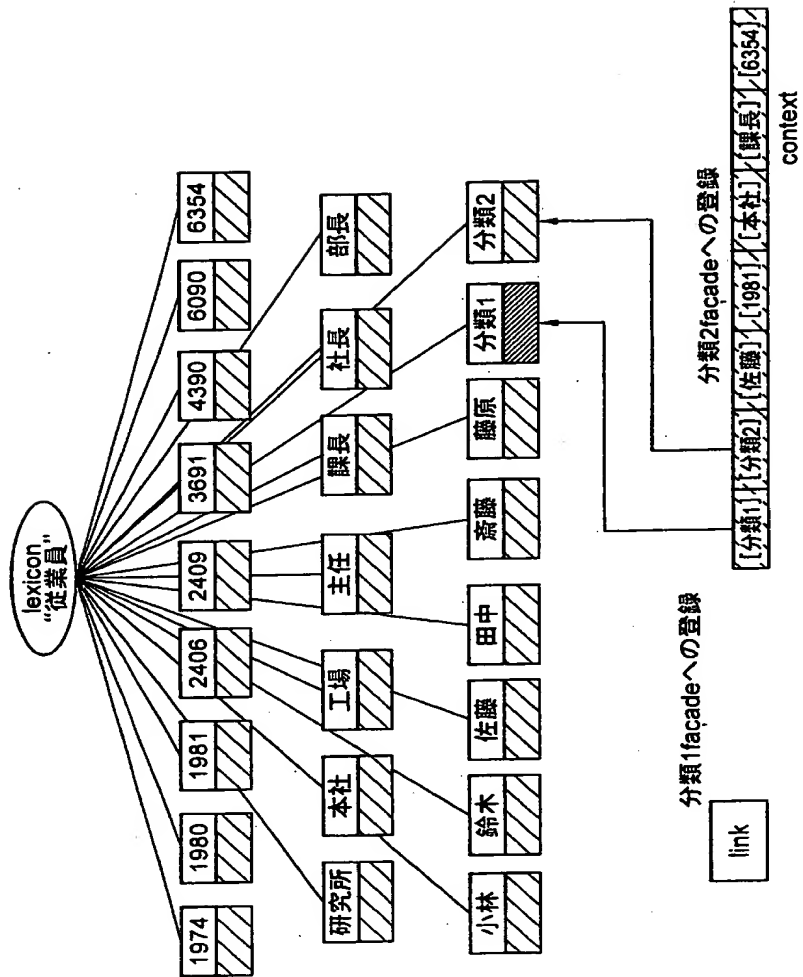


【図 1 0】



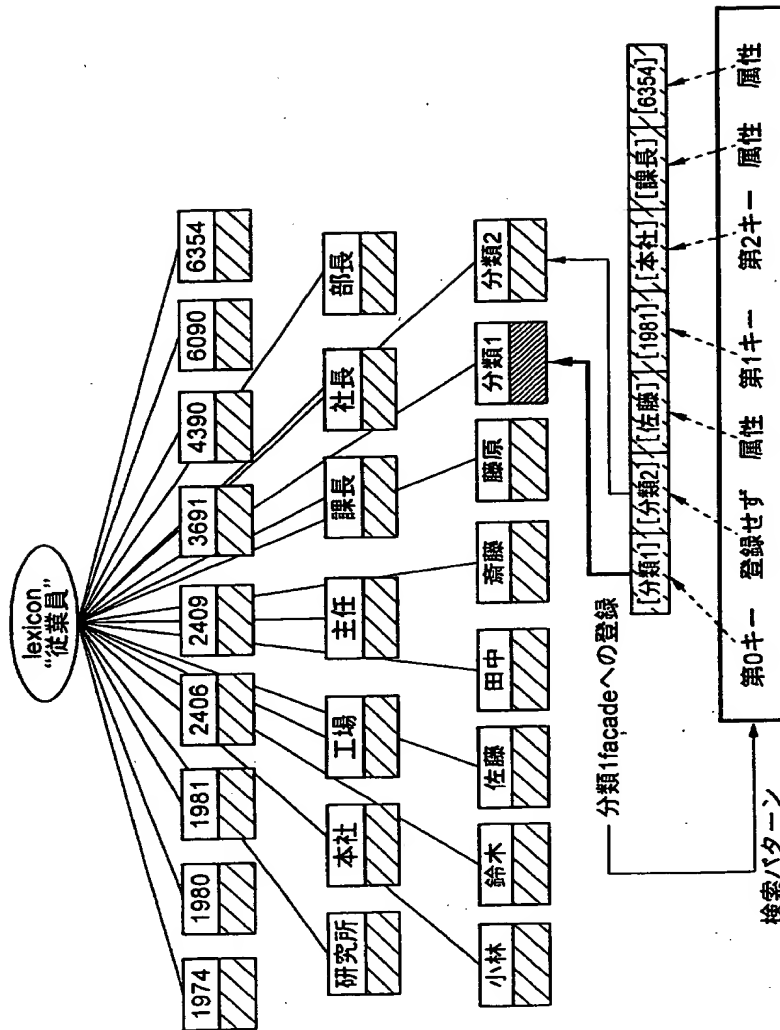
contextのfaçadeへの登録処理: フローチャート

【図 1 1】



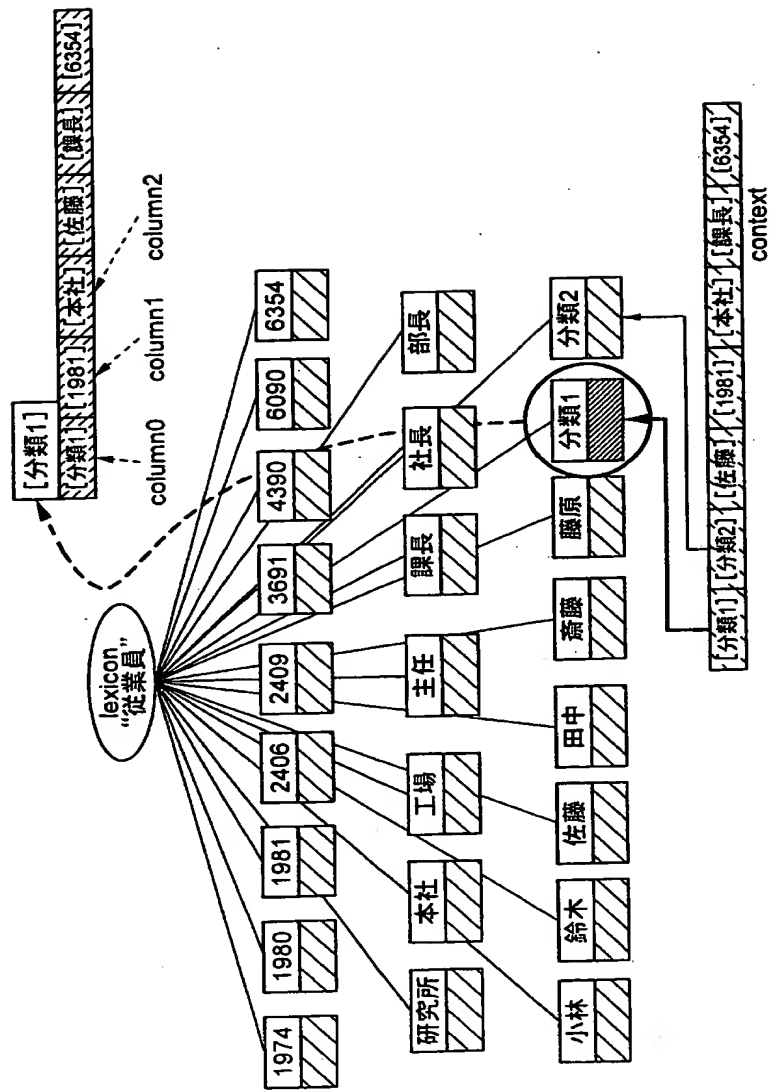
最初のcontextを"分類1"façadeへ登録する

【図 12】



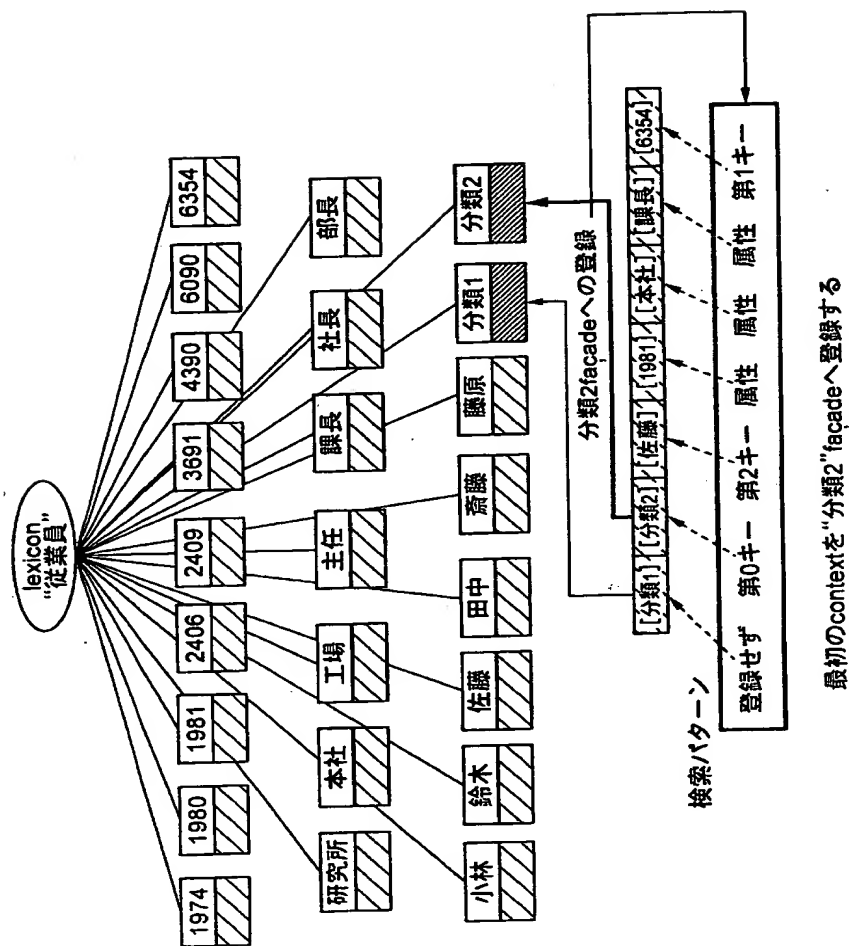
最初のcontextを"分類1"facadeへ登録する時の検索キーの指定の仕方

【図13】

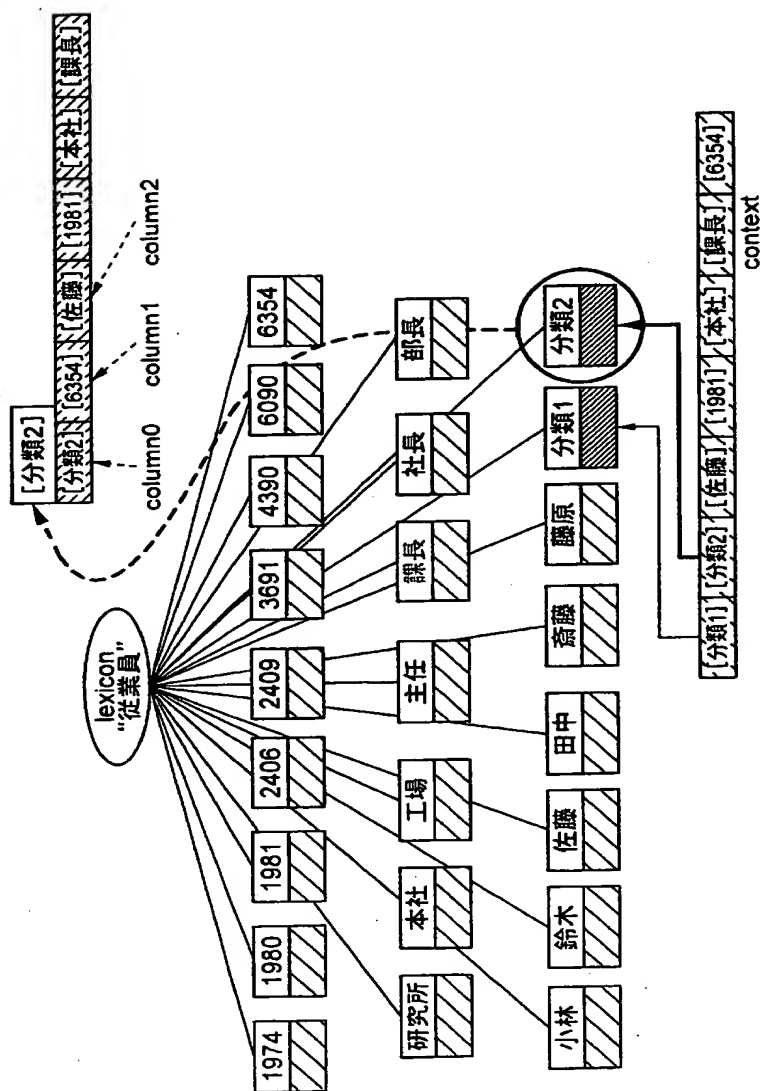


最初のcontextを"分類1"facadeへ登録した時の"分類1"facadeの状態

【図 1 4】

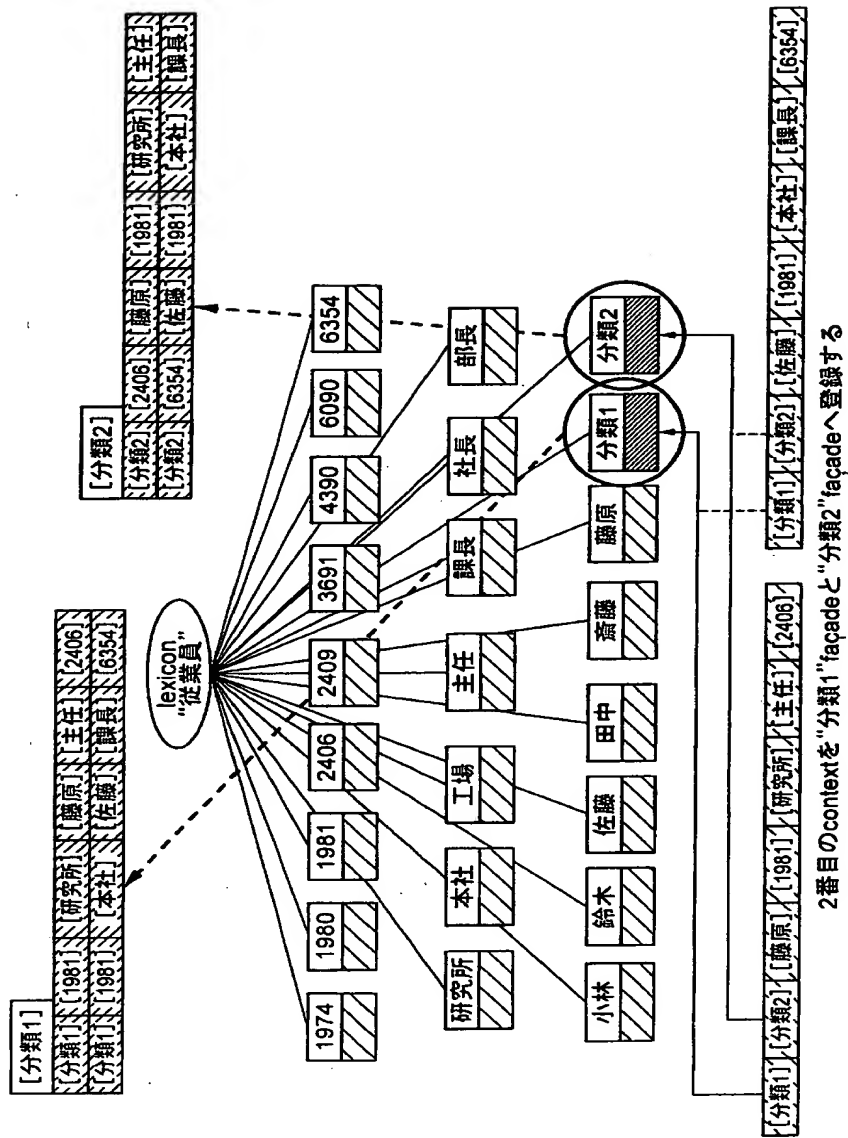


【図15】

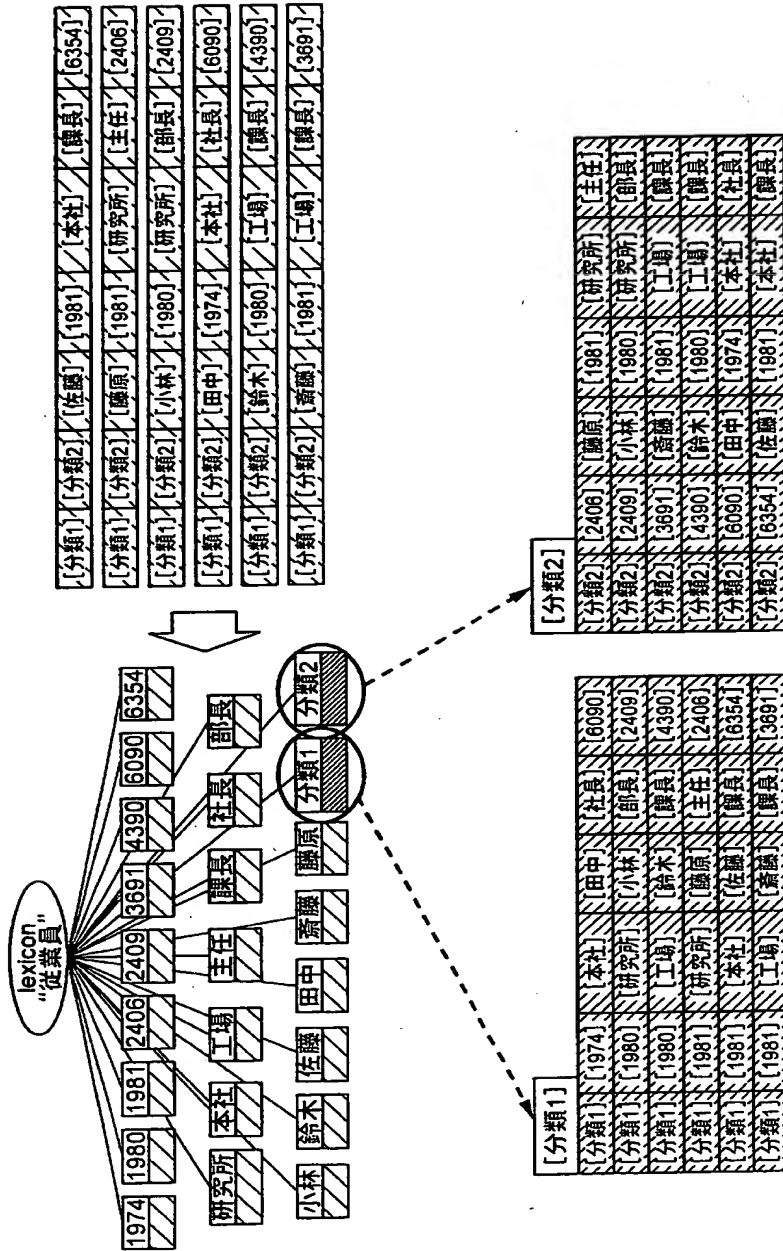


最初のcontextを"分類2"facadeへ登録した時の"分類2"facadeの状態

【図16】

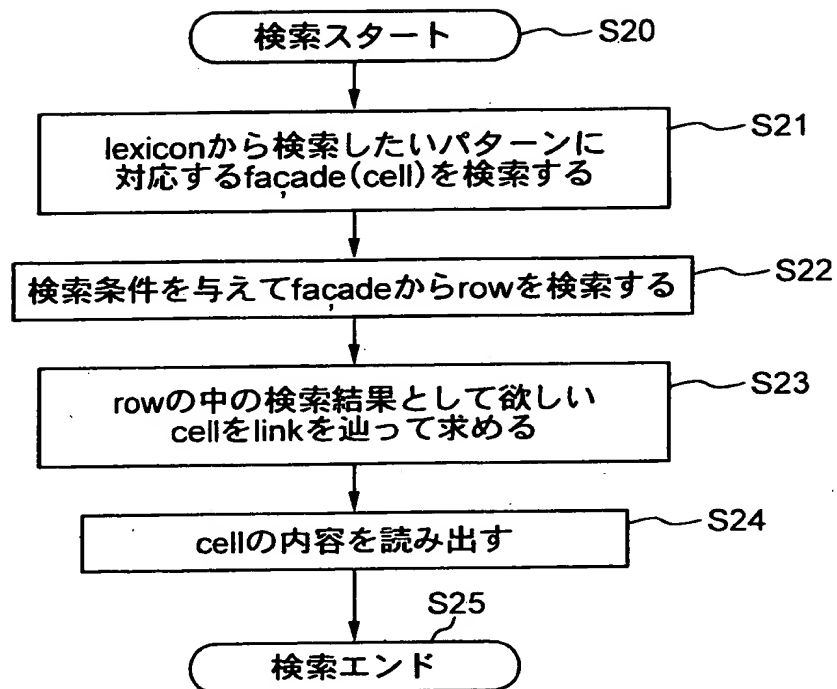


【図 17】

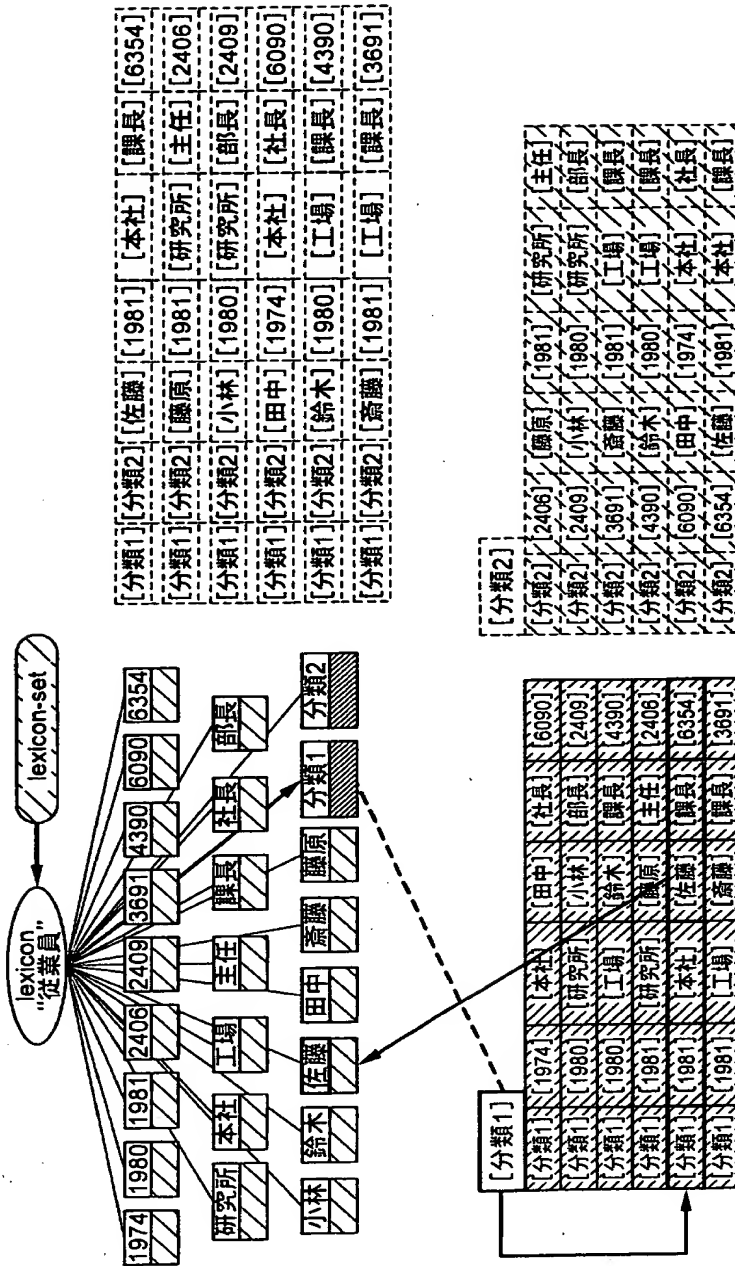


全てのcontextを“分類1”facadeと“分類2”facadeへ登録する

【図 1 8】

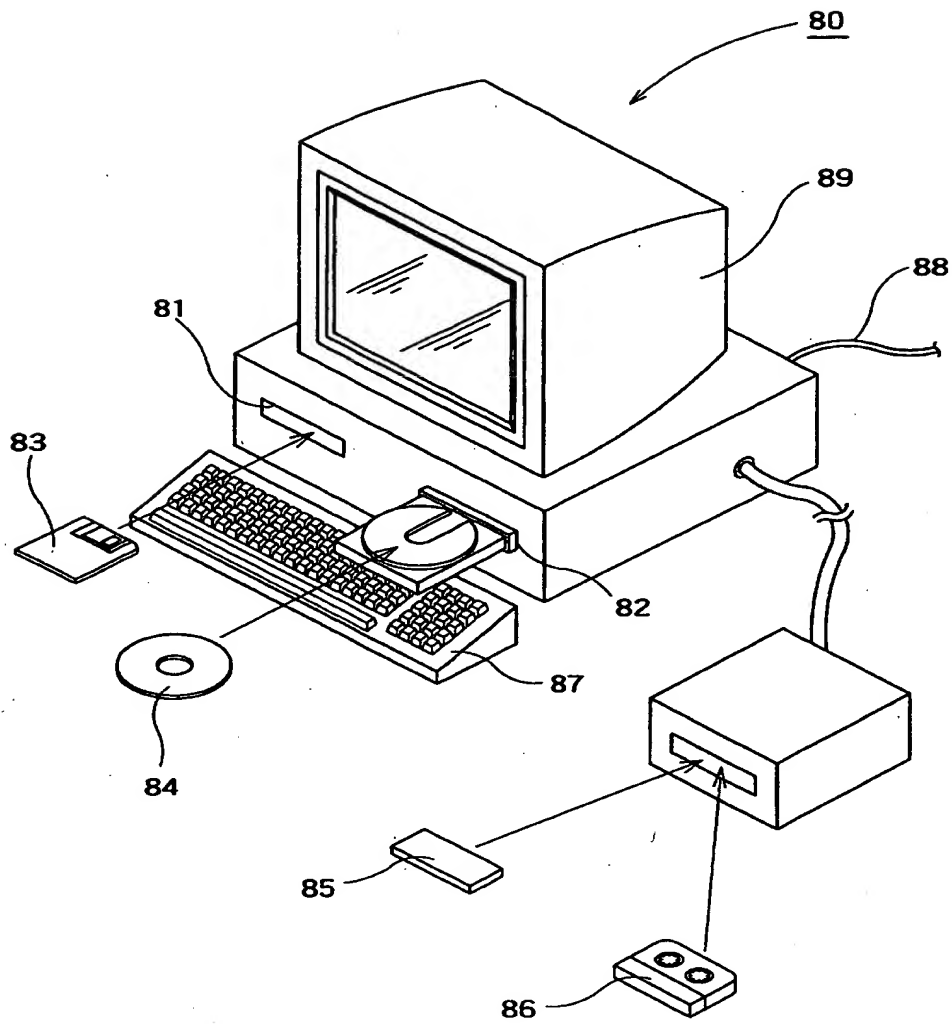


【図 19】

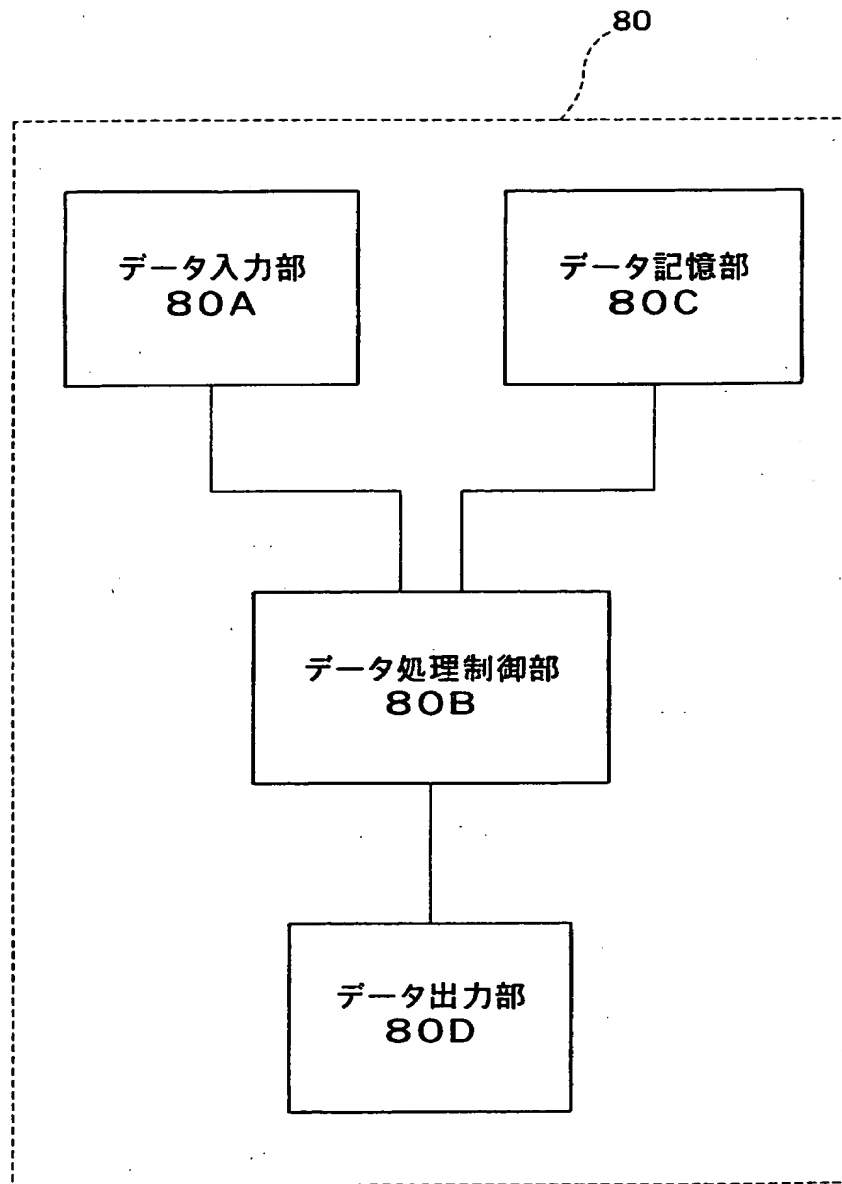


データの管理装置の検索処理

【図 20】



【図 21】



【書類名】 要約書

【要約】

【課題】 データ処理性能を向上させ、各種のデータ構造に対する適応性が高く、データ構造の変更も容易で、大規模システムの開発も容易にするデータ管理方法を提供することを目的とする。

【解決手段】 データの実体を格納するための領域（セル）に、セルに格納されたデータに関連した複数のセルの間の関係をポインタの列で表すデータ（ロウ）を格納する領域（ファサード）を付随させたデータ構造とすることによって、スキーマ無しにデータの実体そのものを管理することができ、データの効率的な操作（登録、更新、削除、検索）を実現することができる。

【選択図】 図2

出 願 人 履 歴 情 報

識別番号 [000003078]

1. 変更年月日 1990年 8月22日
[変更理由] 新規登録
住 所 神奈川県川崎市幸区堀川町72番地
氏 名 株式会社東芝
2. 変更年月日 2001年 7月 2日
[変更理由] 住所変更
住 所 東京都港区芝浦一丁目1番1号
氏 名 株式会社東芝